5 Myhill-Nerode

In this chapter, we consider three additional characterizations of regular languages:

1. Myhill relations,
2. weak Nerode relations,
3. and Nerode relations.

We will show that these three characterizations can be used to characterize regular languages by proving them equivalent to the existence of a (deterministic) finite automaton.

5.1 Definitions

Before we can state the Myhill-Nerode theorem, we need a number of auxiliary definitions. We roughly follow [23].

Definition 5.1.1. *Let $\equiv$ be an equivalence relation. The* ***equivalence class*** *of $u \in \Sigma^*$ w.r.t. $\equiv$ is the set of all v such that $u \equiv v$. It is denoted by $[u]_\equiv$.*

Definition 5.1.2. *Let $\equiv$ be an equivalence relation. $\equiv$ is of* ***finite index*** *if and only if the set of $\{[u]_\equiv \mid u \in \Sigma^*\}$ is finite.*

Due to the lack of native support for quotient types in Coq, we formalize equivalence relations of finite index as surjective functions from Σ^* to a finite type X.

Definition 5.1.3. *Let X be finite. Let $f : \Sigma^* \mapsto X$ be surjective. Let $u, v \in \Sigma^*$. f is an* ***equivalence relation of finite index****. u and v are equivalent w.r.t. f if and only if $f(u) = f(v)$. $f(u)$ is the equivalence class of u w.r.t. f.*

```
Record Fin_Eq_Cls :=
      { fin_type  : finType;
        fin_func :> word -> fin_type;
         fin_surjective  :  surjective  fin_func  }.
```

Definition 5.1.4. *Let f be as above. Let $x \in X$. $w \in \Sigma^*$ is a* ***representative*** *of x if and only if $f(w) = x$. Since f is surjective, every w has a representative. We write $cr(x)$ to denote the* ***canonical representative*** *of x, which we obtain by constructive choice.*

```
Definition cr (f: Fin_Eq_Cls) x := xchoose ( fin_surjective  f x).
```

5.1.1 Myhill Relations

Definition 5.1.5. *Let $\equiv$ be an equivalence relation of finite index. $\equiv$ is a **Myhillrelation** [23] on L if and only if*

*(i) $\equiv$ is **right congruent**, i.e. for all $u, v \in \Sigma^*$ and $a \in \Sigma$,*

$$u \equiv v \Rightarrow u \cdot a \equiv v \cdot a.$$

*(ii) $\equiv$ **refines** L, i.e. for all $u, v \in \Sigma^*$,*

$$u \equiv v \Rightarrow (u \in L \iff v \in L).$$

Myhill relations are commonly referred to as "Myhill-Nerode relations". In this thesis, it makes sense to split the concept of a Myhill relation from that of the Nerode relation. Apart from the Nerode relation, which can be seen as the coarsest Myhill relation, we also define weak Nerode relations that have no direction connection to Myhill relations. Thus, we strictly separate the characterizations.

Mathematically, Myhill relations are required to be of finite index. We only formalize equivalence relations of finite index. Thus, proving that a Myhill relation is of finite index mathematically corresponds to constructing a Myhill relation in our formalization.

```
Definition right_congruent {X} (f: word -> X) :=
        forall u v a, f u = f v -> f (rcons u a) = f (rcons v a).
Definition refines {X} (f: word -> X) :=
        forall u v, f u = f v -> u \in L = (v \in L).
Record Myhill_Rel :=
        { myhill_func :> Fin_Eq_Cls;
          myhill_congruent : right_congruent myhill_func ;
          myhill_refines : refines myhill_func }.
```

Myhill relations correspond to the equivalence relations defined as the pairs of words (u, v) whose runs on a DFA A end in the same state. These relations are right congruent, refine $\mathcal{L}(A)$ and are of finite index as A has finitely many states. We will later give a formal proof of this.

5.1.2 Nerode Relations

Definition 5.1.6. *Let $u, v \in \Sigma^*$. Let L be a language. We define the **Nerode relation** $\doteq_L$ on L such that*

$$u \doteq_L v \iff \forall w \in \Sigma^*.\ uw \in L \Leftrightarrow vw \in L.$$

The Nerode relation given above is a propositional statement in Coq. To proof that the Nerode relation is of finite index, we require an equivalence relation, i.e. a function f from words to a finite type, such that f is equivalent to $\doteq_L$.

```
Definition equiv_suffix {X} (f: word -> X) :=
    forall u v, f u = f v <-> suffix_equal u v.
Record Nerode_Rel :=
    { nerode_func :> Fin_Eq_Cls;
      nerode_equiv: equiv_suffix nerode_func }.
```

Definition 5.1.7. *Let L be a language and let $\equiv$ be an equivalence relation. We say that $\equiv$ is a **weak Nerode relation** on L if and only if*

$$\forall u, v \in \Sigma^*.\ u \equiv v \implies u \doteq_L v.$$

```
Definition suffix_equal u v :=
    forall w, u++w \in L = (v++w \in L).
Definition imply_suffix {X} (f: word -> X) :=
      forall u v, f u = f v -> suffix_equal u v.
Record Weak_Nerode_Rel :=
      { weak_nerode_func :> Fin_Eq_Cls;
        weak_nerode_imply: imply_suffix weak_nerode_func }.
```

It appears that the notion of a weak Nerode relation is not found in the literature. We will later prove them weaker than Myhill relations, in the sense that every Myhill relation is also a weak Nerode relation.

5.1.3 Myhill-Nerode Theorem

We can now move on to the statement of the Myhill-Nerode theorem [23].

Theorem 5.1.8. *(Myhill-Nerode) Let L be a language. The following four statements are equivalent:*

1. *there exists a deterministic finite automaton that accepts L;*
2. *there exists a Myhill relation on L;*
3. *there exists a weak Nerode relation on L;*
4. *the Nerode relation on L is of finite index.*

Our proof of equivalence will have the following structure:

$$\mathit{DFA} \implies \mathit{Myhill} \implies \mathit{weak\ Nerode} \implies \mathit{Nerode} \implies \mathit{DFA}.$$

We will first show a, b, and d. We will then give a proof of c, which is the most challenging proof and formalization in this chapter.

5.2 Finite Automata to Myhill relations

We assume we are given a DFA A. We will be using the states of A as equivalence classes. Our goal is to construct a Myhill relation, for which we will need an equivalence relation of finite index. Therefore, we first need to ensure that the mapping from words to equivalence classes is surjective. Thus, we consider the equivalent, connected automaton $A_c = (Q_c, s_c, F_c, \delta_c)$ (Definition 4.2.1), which has only reachable states. This enables us to construct a surjective function from words to the states of A_c.

Definition 5.2.1. *Let $u \in \Sigma^*$. Let σ be the run of u on A_c. We define $f_M : \Sigma^* \mapsto Q_c$ such that $f_M(U)$ is the last state in σ, i.e.*

$$f_M(u) := \sigma_{|\sigma|-1}.$$

Note that f_M is surjective (follows directly from Lemma 4.2.5) and, thus, an equivalence relation of finite index.

```
Definition f_M := fun w => last (dfa_s A_c) (dfa_run A_c w).
Lemma f_M_surjective: surjective f_M.
Definition f_fin : Fin_Eq_Cls :=
    {| fin_func := f_M;
       fin_surjective := f_M_surjective |}.
```

In order to show that f_M is a Myhill relation, we prove that it fulfills Definition 5.1.5.

Lemma 5.2.2. *f_M is right congruent.*

Proof. Let $u, v \in \Sigma^*$ such that $f_M(u) = f_M(v)$. Let $a \in \Sigma$. Since A is deterministic, we get $f_M(ua) = f_M(va)$. □

Lemma 5.2.3. *f_M refines $\mathcal{L}(A_c)$.*

Proof. Let $u, v \in \Sigma^*$ such that $f_M(u) = f_M(v)$. By definition of f_M, the runs u and v on A end in the same state. Thus, either u and v are both accepted, or both not accepted. □

Theorem 5.2.4. *f_M is a Myhill relation on $\mathcal{L}(A)$.*

Proof. By Lemma 4.2.3, we have $\mathcal{L}(A_c) = \mathcal{L}(A)$. Thus, it suffices to show that f_M is a Myhill relation on $\mathcal{L}(A_c)$. This follows with Lemma 5.2.2 and Lemma 5.2.3. □

We only have extensional equality on $\mathcal{L}(A_c)$ and $\mathcal{L}(A)$ in Coq. Thus, we first show that f_M is a Myhill relation on $\mathcal{L}(A_c)$. Then, we show that we can get a Myhill relation on $\mathcal{L}(A)$ from a Myhill relation on $\mathcal{L}(A_c)$.

```
Definition dfa_to_myhill ' : Myhill_Rel (dfa_lang A_c) :=
    {| myhill_func := f_fin ;
      myhill_congruent := f_M_right_congruent;
       myhill_refines := f_M_refines |}.
Lemma myhill_lang_eq L1 L2: L1 =i L2 -> Myhill_Rel L1 -> Myhill_Rel L2.
Definition dfa_to_myhill : Myhill_Rel (dfa_lang A) :=
    myhill_lang_eq (dfa_connected_correct A) dfa_to_myhill '.
```

This concludes the proof of step (a).

5.3 Myhill relations to weak Nerode relations

We show that, if there exists a Myhill relation, there also exists a weak Nerode relation. In fact, we will prove that any Myhill relation *is* a weak Nerode relation.

Theorem 5.3.1. *Let f be a Myhill relation on a language L. Then f is a weak Nerode relation on L.*

Proof. Let $u, v \in \Sigma^*$ such that $u =_f v$. We have to show that for all $w \in \Sigma^*$, $uw \in L \Leftrightarrow vw \in L$. By induction on w.

1. For $w = \varepsilon$, we get $u \in L \Leftrightarrow v \in L$ as f refines L.
2. For $w = aw'$, we get $ua =_f va$ by congruence of f and thus, by inductive hypothesis, $uaw' \in L \Leftrightarrow vaw' \in L$.

□

```
Lemma myhill_suffix: imply_suffix L f.
Definition myhill_to_weak_nerode: Weak_Nerode_Rel L :=
    {| weak_nerode_func := f;
      weak_nerode_imply := myhill_suffix |}.
```

This concludes step (b) of Theorem 5.1.8.

5.4 Nerode relations to Finite Automata

We prove step (d) of Theorem 5.1.8. If the Nerode relation on a language L is of finite index, we can construct a DFA that accepts L. The construction is very straightforward and uses the equivalence classes of the Nerode relation as the set of states for the automaton.

Definition 5.4.1. *Let L be a language. Let X be a finite type. Let $f : \Sigma^* \mapsto X$ be the equivalence relation which proves that the Nerode relation on L is of finite index. We* *construct DFA A such that*

$$\begin{array}{rcl} s & := & f(\varepsilon) \\ F & := & \{x | x \in X \wedge cr(x) \in L\} \\ \delta & := & \{(x, a, f(cr(x)a)) \mid x \in X, a \in \Sigma\} \\ A & := & (X, s, F, \delta). \end{array}$$

```
Definition nerode_to_dfa :=
    {| dfa_s := f [::];
       dfa_fin := [pred x | cr f x \in L ];
       dfa_step := [fun x a => f (rcons (cr f x) a)] |}.
```

In order to show that A accepts the language L, we first need to connect runs on A to the equivalence classes, i.e. the range of f. The following lemma gives a direct connection.

Lemma 5.4.2. *Let $w \in \Sigma^*$. Let σ be the run of w on A starting in s. We have that the last state of σ is the equivalence class of w, i.e.*

$$\sigma_{|\sigma|-1} = f(w).$$

Proof. We proceed by induction on w from right to left.

1. For $w = \varepsilon$ we have $s = f(\varepsilon)$.

2. For $w = w'a$ we know that the run of w' on A starting in s ends in $f(w')$. It remains to show that $(f(w'), a, f(w)) \in \delta$. We have $cr(f(w'))a =_f w$, i.e. $f(cr(f(w'))a) = f(w)$ by definition of f. Thus, it suffices to show $(f(w'), a, f(cr(f(w'))a)) \in \delta$, which holds by definition of δ.

□

Theorem 5.4.3. *A accepts L, i.e. $\mathcal{L}(A) = L$.*

Proof. Let $w \in \Sigma^*$. Let σ be the run of w on A starting in s. w is accepted if and only if $\sigma_{|\sigma|-1} \in F$, i.e. if and only if $cr(\sigma_{|\sigma|-1}) \in L$. We have $w =_f cr(\sigma_{|\sigma|-1})$ and therefore $w \in L \Leftrightarrow cr(\sigma_{|\sigma|-1}) \in L$. Thus w is accepted if and only if $w \in L$. □

The resulting automaton is minimal, i.e. there exists no other automaton that accepts the same language and has less states than A.

This concludes step d of Theorem 5.1.8.

5.5 Minimizing Equivalence Classes

Finally, we prove that if there is a weak Nerode relation on a language L, the Nerode relation is of finite index. This proves step c of the Theorem 5.1.8 and thus complets its proof. For this purpose, we employ a table-filling algorithm [19] to find indistinguishable states under the Myhill-Nerode relation. This algorithm was originally intended to be used on automata. It identifies (un)distinguishable states based on their successors. We use the finite type X, i.e., the equivalence classes, instead of states.

For the remainder of this section, we assume we are given a language L and a weak Nerode relation f_0.

We employ a fixed-point construction to find equivalence classes that are $\doteq_L$-equivalent. In each step, we add those equivalences classes that are distinguishable based on equivalence classes that were distinguishable in the previous step. The initial set of distinguishable equivalence classes are distinguishable by the inclusion of their canonical representative in L. We denote this initial set $dist_0$.

$$dist_0 := \{(x, y) \in F \times F \mid cr(x) \in L \Leftrightarrow cr(y) \notin L\}.$$

```
Definition distinguishable := [ fun x y => (cr f_0 x) \in L != ((cr f_0 y) \in L) ].
Definition dist0 := [set x | distinguishable x.1 x.2 ].
```

To find more distinguishable equivalence classes, we have to identify equivalence classes that "lead" to distinguishable equivalence classes. In analogy to the minimization procedure on automata, we define successors of equivalence classes with respect to a single character. The intuition is that two states are distinguishable if they are succeeded by a pair of distinguishable states. Conversely, if a pair of states is not distinguishable, then their predecessors can not be distinguished by those states. Thus, two states are undistinguishable if none of their succeeding pairs of states are distinguishable.

Definition 5.5.1. *Let $x, y \in X$ and $a \in \Sigma$. We define $succ_a$ and $psucc_a$. $succ_a(x) := f_0(cr(x) \cdot a)$ and $psucc_a(x, y) := (succ_a(x), succ_a(y))$.*

```
Definition succ := [ fun x a => f_0 ((cr f_0 x) ++ [::a]) ].
Definition psucc := [ fun x y => [ fun a => (succ x a, succ y a) ] ].
```

The fixed-point algorithm tries to extend the set of distinguishable equivalence classes by looking for a pair of equivalence classes that transitions to a pair of distinguishable equivalence classes. Given a set of pairs of equivalence classes *dist*, we define the set of pairs of distinguishable equivalence classes that have successors in *dist* as

$$dist_S(dist) := \{(x, y) \mid \exists a.\, psucc_a(x, y) \in dist\}.$$

```
Definition distS (dist : {set X*X}) :=
    [set (x,y) | x in X, y in X & [ exists a, psucc x y a \in dist ] ].
```

Definition 5.5.2. *Let dist be a subset of $X \times X$. We define one-step-dist such that*

$$\textit{one-step-dist}(dist) := dist_0 \cup dist \cup distinct_S(dist).$$

Definition one_step_dist dist := dist0 :|: dist :|: (distS dist).

Lemma 5.5.3. *one-step-dist is monotone and has a fixed-point.*

Proof. Monotonicity follows directly from the monotonicity of $\cup$. The number of sets in $X \times X$ is finite. Therefore, *one-step-dist* has a fixed point. We iterate *one-step-dist* $|X * X| + 1 = |X|^2 + 1$ times on the empty set. Since there can only ever be $|X * X|$ items in the result of *one-step-dist*, we will find the fixed point in no more than $|X * X| + 1$ steps.

Let ***distinct*** be the fixed point of *one-step-dist* and let it be denoted by $\ncong$. We write ***equiv*** for the complement of *distinct* and denote it $\cong$. □

Definition lfp := iter #|T|.+1 F set0.

Definition distinct := lfp one_step_dist .

We now show that $\cong$ is equivalent to the Nerode relation. Formally, this means constructing a function that fulfills our definition of an equivalence relation of finite index. Then, we prove that this equivalence relation is equivalent to the the Nerode relation. First, we will prove that $\cong$ is an equivalence relation. Then, we will prove it equivalent to the Nerode relation in two separate steps, since the two directions require different strategies.

Lemma 5.5.4. $\cong$ *is an equivalence relation.*

Proof. We first state transitivity of $\cong$ in terms of $\ncong$:

$$\forall x, y, z \in X. \neg(x \ncong y) \implies \neg(y \ncong z) \implies \neg(x \ncong z). \tag{*}$$

It suffices to show that *distinct* is anti-reflexive, symmetric and fulfills (*). Note that complementary transitivity, anti-reflexivity and symmetry are closed under union. We proceed by fixed-point induction.

1. For $\textit{one-step-dist}(\textit{dist}) = \emptyset$ we have anti-reflexivity, symmetry and (*) by the properties of $\emptyset$.

2. For $\textit{one-step-dist}(\textit{dist}) = \textit{dist}'$ we have *dist* anti-reflexive, symmetric and (*). It remains to show that $\textit{dist}_0$ and $\textit{distinct}_S(\textit{dist})$ are anti-reflexive, symmetric and fulfill (*).

   $\textit{dist}_0$ is anti-reflexive, symmetric and fulfills (*) by definition.

   $\textit{distinct}_S(\textit{dist})$ can be seen as an intersection of a symmetric subset of $X \times X$ defined by $\textit{psucc}_a$ and *dist*, the latter of which is anti-reflexive, symmetric and fulfills (*). Thus, $\textit{distinct}_S(\textit{dist})$ is anti-reflexive, symmetric and fulfills (*).

   Therefore, $\textit{dist}'$ is anti-reflexive, symmetric and fulfills (*).

□

Lemma equiv_refl x: x ~= x.

Lemma equiv_sym x y: x ~= y -> y ~= x.

Lemma equiv_trans x y z: x ~= y -> y ~= z -> x ~= z.

Lemma 5.5.5. *Let* $u, v \in \Sigma^*$. $f_0(u) \cong f_0(v) \implies u \doteq_L v$.

Proof. Let $w \in \Sigma^*$. We then show the contraposition of the claim:

$$uw \in L \not\Leftrightarrow vw \in L \implies f_0(u) \not\cong f_0(v).$$

By induction on w.

1. For $w = \varepsilon$ we have $u \in L \not\Leftrightarrow v \in L$ which gives us $(f_0(u), f_0(v)) \in dist_0$. Thus, $f_0(u) \not\cong f_0(v)$.
2. For $w = aw'$ we have $uaw \in L \not\Leftrightarrow vaw \in L$. We have to show $f_0(u) \not\cong f_0(v)$, i.e. $(f_0(u), f_0(v)) \in distinct$. By definition of *distinct*, it suffices to show $(f_0(u), f_0(v)) \in one\text{-}step\text{-}dist(distinct)$.

   For this, we prove $(f_0(u), f_0(v)) \in distinct_S(distinct)$. By $uaw \in L \not\Leftrightarrow vaw \in L$ we have $(f_0(cr(u)a), f_0(cr(v)a)) \in dist_0$.

   It remains to show that $f_0(cr(u)a) \not\cong f_0(cr(v)a)$ which we get by inductive hypothesis. For this, we need to show that $cr(u)aw \in L \not\Leftrightarrow cr(v)aw \in L$.

   By the properties of f, we get $cr(u)aw \in L \Leftrightarrow uaw \in L$ and $cr(v)aw \in L \Leftrightarrow vaw \in L$. Thus, $cr(u)aw \in L \not\Leftrightarrow cr(v)aw$.

□

Lemma 5.5.6. *Let* $u, v \in \Sigma^*$. *If* $f_0(u) \not\cong f_0(v)$, *then* u *and* v *are* ***not*** *equivalent wr.t. the Merode relation, i.e.* $f_0(u) \not\cong f_0(v) \implies u \neq_L v$.

Proof. We do a fixed-point induction.

1. For $one\text{-}step\text{-}dist(dist) = \emptyset$ we have $(f_0(u), f_0(v)) \in \emptyset$ and thus a contradiction.
2. For $one\text{-}step\text{-}dist(dist) = dist'$ we have $(f_0(u), f_0(v)) \in dist'$. We do a case distinction on $dist'$.
   a) $(f_0(u), f_0(v)) \in dist_0$. We have $u \in L \not\Leftrightarrow v \in L$. Thus, $u \neq_L v$ as witnessed by $w = \varepsilon$.
   b) $(f_0(u), f_0(v)) \in dist$. By inductive hypothesis, $u \neq_L v$.
   c) $(f_0(u), f_0(v)) \in distinct_S(dist)$. We have $a \in \Sigma$ with $psucc_a(f_0(u), f_0(v))) \in dist$. By inductive hypothesis, we get $cr(u)a \neq_L cr(v)a$ as witnessed by $w \in \Sigma^*$ such that $cr(u)aw \in L \not\Leftrightarrow cr(v)aw \in L$.

      By the properties of f, we get $cr(u)aw \in L \Leftrightarrow uaw \in L$ and $cr(v)aw \in L \Leftrightarrow vaw \in L$. Thus, we have $u \neq_L v$ as witnessed by aw.

□

Corollary 5.5.7. *Let $u, v \in \Sigma^*$. We have that*

$$f_0(u) \cong f_0(v) \iff u \doteq_L v.$$

Proof. Follows immediately with Lemma 5.5.5 and Lemma 5.5.6. □

```
Lemma equiv_suffix_equal u v: u ~=_f_0 v -> suffix_equal L u v.
Lemma distinct_not_suffix_equal u v:
      u ~!=_f_0 v ->
      exists w, u ++ w \in L != (v ++ w \in L).
Lemma equivP u v:
       reflect ( suffix_equal L u v)
               (u ~=_f_0 v).
```

Definition 5.5.8. *Let $w \in \Sigma^*$. We define*

$$f_{min}(w) := \{x \mid x \in X,\ f_0(w) \cong x\}.$$

Note that the range of f_{min} is finite (since X is finite) and does not contain the empty set (due to reflexivity of $\cong$).

Lemma 5.5.9. *f_{min} is surjective.*

Proof. Let $s \in range(f_{min})$. Since $s \neq \emptyset$, there exists $x \in X$ such that $x \in s$. We have $f_0(x) = f_0(cr(x))$ and therefore $f_0(x) \cong f_0(cr(x))$ by reflexivity of $\cong$. Thus, $f_0(cr(x)) = s$ since $f_{min}(x) = f_{min}(cr(x)) = s$. □

Lemma 5.5.10. *For all $u, v \in \Sigma^*$ we we have*

$$f_{min}(u) = f_{min}(v) \iff f_0(u) \cong f_0(v).$$

Proof. "$\Rightarrow$" We have $f_{min}(u) = f_{min}(v)$ and thus $f_0(u) \cong f_0(v)$.

"$\Leftarrow$" We have $f_0(u) \cong f_0(v)$. Let $x \in X$. It suffices to show that $f_0(u) \cong x$ if and only if $f_0(v) \cong x$. This follows with symmetry and transitivity of $\cong$. □

Lemma 5.5.11. *f_{min} is equivalent to the Nerode relation, i.e. f_{min} is surjective and for all $u, v \in \Sigma^*$ we have*

$$f_{min}(u) = f_{min}(v) \iff u \doteq_L v.$$

Proof. We have proven surjectivity in Lemma 5.5.9. By Lemma 5.5.10 we have $f_{min}(u) = f_{min}(v)$ if and only if $f_0(u) \cong f_0(v)$. By corollary 5.5.7 we have $f_0(u) \cong f_0(v)$ if and only if $u \doteq_L v$. Thus, $f_{min}(u) = f_{min}(v)$ if and only if $u \doteq_L v$. □

The formalization of f_{min} is slightly more involved than the mathematical construction. We first need to define the finite type of f_{min}'s range, which we do by enumerating all possible values of f_{min}.

```
Definition equiv_repr x := [set y | x ~= y].
Definition X_min := map equiv_repr (enum (fin_type f_0 )).
Definition f_min w := SeqSub (equiv_repr_mem (f_0 w)).
```

We then prove lemmas 5.5.9, 5.5.10 and Theorem 5.5.11 which are consequential and straight-forward.

```
Lemma f_min_surjective: surjective f_min.
Lemma f_minP u v:
    reflect (f_min u = f_min v)
            (u ~=_f_0 v).
Lemma f_min_correct: equiv_suffix L f_min.
Definition f_min_fin : Fin_Eq_Cls :=
   {| fin_surjective := f_min_surjective |}.
```

We can now state the result of this section.

Corollary 5.5.12. *The Nerode relation is of finite index.*

Proof. This follows directly from Lemma 5.5.4 and Lemma 5.5.11. □

```
Definition weak_nerode_to_nerode: Nerode_Rel L :=
   {| nerode_func := f_min_fin ;
      nerode_equiv := f_min_correct |}.
```

This concludes step ⓒ of Theorem 5.1.8 and, thus, this chapter. The characterizations presented in this chapter are very compact, mathematically. They also lend themselves very well to formalization.