Chapter 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 concepts can be used to characterize regular languages by proving them equivalent to the existence of a (deterministic) finite automaton. Our proof of equivalence will have the following structure:

$$\underbrace{\mathit{DFA} \implies \rlap{$\overbrace{\phantom{\mathit{Myhill} \implies \mathit{weak\ Nerode}}}^{\textcircled{b}}$}\mathit{Myhill}}_{\textcircled{a}} \implies \underbrace{\mathit{weak\ Nerode} \implies \rlap{$\overbrace{\phantom{\mathit{Nerode} \implies \mathit{DFA}}}^{\textcircled{d}}$}\mathit{Nerode}}_{\textcircled{c}} \implies \mathit{DFA}.$$

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

5.1 Definition

The following definitions (roughly following [23]) will lead us to the statement of the Myhill-Nerode theorem.

Definition 5.1.1. *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. *$\equiv$ 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 $f : \Sigma^* \mapsto X$ be surjective. The relation $\equiv_f$ is defined such that for all $u, v \in \Sigma^*$*

$$u \equiv_f v \iff f(u) = f(v).$$

For all $w \in \Sigma^*$, $f(w)$ can be seen as an equivalence class of $\equiv_f$.

It is easy to see that $\equiv_f$ is an equivalence relation. Furthermore, from the finiteness of X, it follows that $\equiv_f$ is of finite index.

```
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 $f(w) = x$. 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. $\equiv$ is a* ***Myhill***[1] ***relation*** *[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 $u, v \in \Sigma^*$,*

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

(iii) $\equiv$ is of ***finite index****.*

[1] 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.

Building on our formalization of equivalence relations of finite index, we only need to give formalizations of (i) and (ii).

```
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 <-> equal_suffix u v.
Record Nerode_Rel :=
      { nerode_func :> Fin_Eq_Cls;
        nerode_equiv: equiv_suffix nerode_func }.
```

Definition 5.1.7. *Let $\equiv$ be an equivalence relation. Let L be a language. 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 equal_suffix 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 -> equal_suffix u v.
Record Weak_Nerode_Rel :=
        { weak_nerode_func :> Fin_Eq_Cls;
          weak_nerode_imply: imply_suffix weak_nerode_func }.
```

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.*

5.2 Minimizing Equivalence Classes

We will prove that if there is a weak Nerode relation on a language L, the Nerode relation is of finite index. This proves step (3) $\implies$ (4) and thus step ⓒ of our plan. For this purpose, we employ a table-filling algorithm [18] to find indistinguishable states under the Myhill-Nerode relation. However, we do not rely on an automaton, as is usually done. In fact, 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 equivalent in the sense of $\doteq_L$. In each step, we add those equivalences classes that are distinguishable based on previously distinguishable equivalence classes. 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 distinct0 := [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 allusion to the minimization procedure on automata, we define successors of equivalence classes. The intuition is that two states are distinguishable if they are succeeded by 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, i.e. equivalent, if none of their succeeding pairs of states are distinguishable.

Definition 5.2.1. *Let $x, y \in X$ and $a \in \Sigma$. We define $succ_a$ and $psucc_a$. $succ_a(x) := cr(x) \cdot a$ and $psucc_a(x, y) := (succ_a(x), succ_a(y))$.*

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 are succeeded by pairs in *dist* as

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

Definition 5.2.2. *Let dist be a subset of $X \times X$. We define unnamed such that*

$$unnamed(dist) := dist_0 \cup dist \cup distinct_S(dist).$$

```
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) ] ].
Definition distinctS ( distinct : {set X*X}) :=
        [set (x,y) | x in X, y in X & [ exists a, psucc x y a \in distinct ] ].
Definition unnamed distinct :=
          distinct0 :|: distinct :|: ( distinctS distinct ).
```

Lemma 5.2.3. *unnamed 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, *unnamed* has a fixed point. We iterate *unnamed* $|X * X| + 1 = |X|^2 + 1$ times on the empty set. Since there can only ever be $|X * X|$ items in the result of *unnamed*, we will find the fixed point in no more than $|X * X| + 1$ steps.

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

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 5.2.4. $\cong$ *is an equivalence relation.*

Proof. We first state transitivity of $\cong$ in terms of $\ncong$ and call this property of $\ncong$ **complementary transitivity**:

$$\forall x, y, z \in X.\ \neg(x \ncong y) \implies \neg(y \ncong z) \implies \neg(x \ncong z).$$

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

1. For $\mathit{unnamed}(\mathit{dist}) = \emptyset$ we have anti-reflexivity, symmetry and complementary transitivity by the properties of $\emptyset$.

2. For $\mathit{unnamed}(\mathit{dist}) = \mathit{dist}'$ we have *dist* anti-reflexive, symmetric and complementarily transitive. It remains to show that dist_0 and $\mathit{distinct}_S(\mathit{dist})$ are anti-reflexive, symmetric and complementarily transitive.

   dist_0 is anti-reflexive, symmetric and complementarily transitive by definition.

   $\mathit{distinct}_S(\mathit{dist})$ can be seen as an intersection of a symmetric subset of $X \times X$ defined by psucc_a and the anti-reflexive, symmetric and complementarily transitive *dist*. Thus, $\mathit{distinct}_S(\mathit{dist})$ is anti-reflexive, symmetric and complementarily transitive.

   Therefore, dist' is anti-reflexive, symmetric and complementarily transitive.

□

```
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 5.2.5. *Let* $u, v \in \Sigma^*$. $u \cong_{f_0} v \implies u \doteq_L v$.

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

$$uw \in L \not\Leftrightarrow vw \in L \implies u \ncong_{f_0} v.$$

Induction on w and generalize over u and v.

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, $u \ncong_{f_0} v$.

2. For $w = aw'$ we have $uaw \in L \not\Leftrightarrow vaw \in L$. We have to show $u \ncong_{f_0} v$, i.e. $(f_0(u), f_0(v)) \in distinct$. By definition of $distinct$, it suffices to show $(f_0(u), f_0(v)) \in unnamed(distinct)$.

   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 $cr(u)a \ncong_{f_0} cr(v)a$ 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 5.2.6. *Let $u, v \in \Sigma^*$. If $u \ncong_{f_0} v$, then u and v are **not** equivalent wr.t. the Merode relation, i.e. $u \ncong_{f_0} v \implies u \not\doteq_L v$.*

Proof. We do a fixed-point induction.

1. For $unnamed(dist) = \emptyset$ we have $(f_0(u), f_0(v)) \in \emptyset$ and thus a contradiction.

2. For $unnamed(dist) = dist'$ 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, $u \not\doteq_L v$ as witnessed by $w = \varepsilon$.

   (b) $(f_0(u), f_0(v)) \in dist$. By inductive hypothesis, $u \not\doteq_L v$.

   (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 $cr(u)a \not\doteq_L cr(v)a$ 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 $u \not\doteq_L v$ as witnessed by aw.

□

Corollary 5.2.7. *Let $u, v \in \Sigma^*$. We have that*

$$u \cong_{f_0} v \iff u \doteq_L v.$$

```
Lemma equiv_equal_suffix u v: u ~=_f_0 v -> equal_suffix L u v.
Lemma distinct_not_equal_suffix u v:
    u ~!=_f_0 v ->
    exists w, u ++ w \in L != (v ++ w \in L).
```

```
Lemma equivP u v:
    reflect ( equal_suffix L u v)
            (u ~=_f_0 v).
```

Definition 5.2.8. *Let $w \in \Sigma^*$. We define*

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

Note that the domain of f_{min} is finite (since X is finite) and contains no empty sets (due to reflexivity of $\cong$).

Lemma 5.2.9. *f_{min} is surjective.*

Proof. Let $s \in dom(f_{min})$. There exists $x \in X$ such that $x \in s$ since $s \neq \emptyset$. We have $f_0(x) = f_0(cr(x))$ and therefore $x \cong_{f_0} cr(x)$ by reflexivity of $\cong$. Thus, $cr(x)$ is a representative of s since $f_{min}(x) = f_{min}(cr(x)) = s$. □

Lemma 5.2.10. *For all $u, v \in \Sigma^*$ we we have*

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

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

"⇐" We have $u \cong_{f_0} v$. 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 from symmetry and transitivity of $\cong$. □

Lemma 5.2.11. *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 5.2.9. By lemma 5.2.10 we have $f_{min}(u) = f_{min}(v)$ if and only if $u \cong_{f_0} v$. By corollary 5.2.7 we have $u \cong_{f_0} v$ 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 domain, 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 5.2.9, 5.2.10 and theorem 5.2.11 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 chapter.

Corollary 5.2.12. *The Nerode relation is of finite index.*

Proof. This follows directly from lemma 5.2.4 and lemma 5.2.11. □

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

This concludes step (3) $\implies$ (4) of theorem 5.1.8 (step ⓒ).

5.3 Finite Automata, Myhill relations, and Nerode relations

First, we will show that, given a DFA A, we can construct a Myhill relation on $\mathcal{L}(A)$. We will then show that this also enables us to construct a weak Nerode relation $\mathcal{L}(A)$. As we have already covered the minimization of equivalence classes in the previous chapter, this proves that the Nerode relation is of finite index.

Conversely, knowing that the Nerode relation on language L is of finite index, we will show that we can construct a DFA that accepts L.

5.3.1 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 5.3.1. *Let $u \in \Sigma^*$. Let σ be the run of u on A_c. We define $f : \Sigma^* \mapsto Q_c$ 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.

```
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 5.3.2. *f_M is right congruent.*

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

Lemma 5.3.3. *f_M refines $\mathcal{L}(A_c)$.*

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

Theorem 5.3.4. *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 from lemma 5.3.2 and lemma 5.3.3. □

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 (1) $\implies$ (2) (ⓐ).

5.3.2 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 5.3.5. *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 $w \in \Sigma^*$, $uw \in L \Leftrightarrow vw \in L$. Induction on w and generalize over u and v.

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 $(2) \implies (3)$ (ⓑ).

5.3.3 Nerode relations to Finite Automata

Finally, we will prove the last step 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 straight-forward and uses the equivalence classes of the Nerode relation as the set of states for the automaton.

Definition 5.3.6. *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*

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

```
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 domain of f. The following lemma gives a direct connection.

Lemma 5.3.7. *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 5.3.8. *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 $w \in L$. □

This concludes the last step of theorem 5.1.8, namely (4) $\implies$ (1) ($\textcircled{d}$), and, consequently, this chapter.