Generalized Finite Automata

Generalized finite automata theory with an application to a decision problem of second-order logic by J.W. Thacher and J.B. Wright
Motivation

- Generalized Finite Automata

- We’re going to use this model to obtain positive solutions to a decision problem of second-order-logic

- Specifically, we’re interested in “**weak monadic second-order logic with k successors**”
Weak Monadic Second-order With K Successors

Explanation of the name:

- Second-order logic: we can quantify not only over elements but also overs sets and relations.
- Weak: the sets we quantify over are finite.
- K successors: the domain is an infinite chain, where each elements has K successors.
Generalized Finite Automata

- Commonly used definition of automaton:
  \[ \mathcal{A} = \langle A, M, a_0, A_F \rangle \]
  - \( A \) is a finite set of states
  - \( M : \Sigma \times A \rightarrow A \) is the direct transition function.
  - \( a_0 \in A \) is the initial state
  - \( A_F \subseteq A \) is the set of final states.

- The behavior of \( \mathcal{A} \) is \( \{ x | M(x) \in A_F \} \) (the strings accepted by \( \mathcal{A} \)).
Generalized Finite Automata

The generalization:

- $\mathcal{A} = \langle A, \alpha_{\sigma_1}, \ldots, \alpha_{\sigma_n}, a_0 \rangle$ ($\Sigma = \{\sigma_1, \ldots, \sigma_n\}$)
- For each $\sigma \in \Sigma$: $\alpha_\sigma : A \rightarrow A$
  \[ \alpha_\sigma(a) = M(\sigma, a) \]
- $bh_{\mathcal{A}}(A_F) = \{x | M(x) \in A_F\}$

(we got rid of the $A_F$ and $\Sigma$ to get a structure that is described as a monadic algebra)
Species

Definition
Species $S = \langle \Sigma, \sigma \rangle$ is defined as follows:

- $\Sigma$ is a set of function symbols
- $\sigma$ is a map from $\Sigma$ into $\mathbb{N}$ (non-negative integers)
- For $f \in \Sigma$, $\sigma(f)$ is the rank of $f$.
- $S$ is finite if $\Sigma$ is finite – the only case we’re interested in.

Notation
- $\Sigma_r = \sigma^{-1}(r)$ : the set of function symbols or rank $r$.
- $\Sigma_0$ : the set of constant symbols of $\Sigma$.
  We always assume $\Sigma_0 \neq \emptyset$. 
\[\Sigma\text{-Algebras of species } \langle \Sigma, \sigma \rangle\]

- **\(\Sigma\text{-Algebra of } S = \langle \Sigma, \sigma \rangle :\)**
  A pair \(\mathcal{A} = \langle A, \alpha \rangle\), where:
  - \(A\) is a set called the *carrier of* \(\mathcal{A}\),
  - \(\alpha\) is a function from \(\Sigma\) into the class of operations on \(A\),
  - \(\alpha(f) = \alpha_f\) is a \(\sigma(f)\)-place function: \(\alpha_f \in A^{A^{\sigma(f)}}\)
  - The 0-place functions \((\alpha(\lambda) \text{ for } \lambda \in \Sigma_0)\) are called constants.

- **Non-deterministic algebra of Species \(\langle \Sigma, \sigma \rangle\):**
  A pair \(\mathcal{R} = \langle R, \rho \rangle\) where \(\rho(f) \in R^{\sigma(f)+1}\)

- **\(T_\Sigma\) : a set of terms uniquely determined by a Species \(\langle \Sigma, \sigma \rangle\),**
  defined as the least subset of \(\Sigma^*\) satisfying:
  - \(\Sigma_0 \subseteq T_\Sigma\)
  - If \(f \in \Sigma_n\) and \(t_1, \ldots, t_n \in T_\Sigma\), then \(ft_1 \ldots t_n \in T_\Sigma\)
**Σ-Automaton of species** \( \langle \Sigma, \sigma \rangle \)

- **Deterministic** \( \mathcal{A} = \langle A, \alpha \rangle \):
  - \( A \) is the set of states
  - For each \( \lambda \in \Sigma_n, \alpha_f \) is the *transition function* for the input symbol \( f \).

- **Non-deterministic** \( \mathcal{R} = \langle R, \rho \rangle \):
  - \( \rho_\lambda \) is a set of initial states corresponding to the individual symbol \( \lambda \).
  - \( \rho_f \) is the *transition relation* for the input \( f \).

** a [non-deterministic] Σ-Automaton of Species S is a finite [non-deterministic] algebra of that species.**
\[\Sigma\text{-Automaton of species }\langle \Sigma, \sigma \rangle\]

**Input:** \(t \in T_\Sigma:\)

- **Deterministic** \(\mathcal{A} = \langle A, \alpha \rangle:\)
  - Output: \(h_\mathcal{A}(t) \in A\) defined as follows:
    - For \(t = \lambda \in \Sigma_0,\) \(h_\mathcal{A}(\lambda) = \alpha_\lambda\)
    - For \(t = f \in \Sigma_n,\) and \(t_1, \ldots, t_n \in T_\Sigma:\)
      \[h_\mathcal{A}(ft_1 \ldots t_n) = \alpha_f(h_\mathcal{A}(t_1), \ldots, h_\mathcal{A}(t_n))\]

- **Non-deterministic** \(\mathcal{R} = \langle R, \rho \rangle:\)
  - Output:
    - For \(t = \lambda \in \Sigma_0,\) \(h_\mathcal{R}(\lambda) = \rho_\lambda\)
    - For \(t = f \in \Sigma_n,\) and \(t_1, \ldots, t_n \in T_\Sigma:\)
      \[h_\mathcal{R}(ft_1 \ldots t_n) = \{a \mid \exists a_1 \ldots \exists a_n[\rho_f(a_1, \ldots, a_n, a) \land \bigwedge_i a_i \in h_\mathcal{R}(t_i)]\}\]
Behavior of $\Sigma$-Automaton

- Given: $\mathcal{A} = \langle A, \alpha \rangle$ (deterministic), $A_F \subseteq A$:
  \[ bh_\mathcal{A}(A_F) = \{ t \mid h_\mathcal{A}(t) \in A_F \} \]

- Given: $\mathcal{R} = \langle R, \rho \rangle$, $R_F \subseteq R$
  \[ bh_\mathcal{R}(R_F) = \{ t \mid h_\mathcal{R}(t) \cap A_F \neq \emptyset \} \]

A set $U \subseteq T_\Sigma$ is **recognizable** if there exists a $\Sigma$-Automaton $\mathcal{A}$ (deterministic or non-deterministic) and set $A_F$ of final states such that $bh_\mathcal{A}(A_F) = U$. 
Equivalence of non-deterministic and deterministic $\Sigma$-automata

*Theorem*: A set $U \subseteq T_\Sigma$ is **recognizable** by a non-deterministic $\Sigma$-Automaton if and only if $U$ is recognizable by a non-deterministic $\Sigma$-Automaton.
Equivalence of non-deterministic and deterministic $\Sigma$-automata

**Proof.** One direction of this equivalence is trivial because any deterministic automaton $\mathcal{A} = \langle A, \alpha \rangle$ corresponds to a nondeterministic automaton $\mathcal{A} = \langle A, \alpha' \rangle$ where $\alpha'_f$ is the graph of $\alpha_f(\alpha'_f(a_1, \ldots, a_n, a) \leftrightarrow \alpha_f(a_1, \ldots, a_n) = a)$.

For any $A_F \subseteq A$,

$$bh_{\mathcal{A}}(A_F) = bh_{\mathcal{A}'}(A_F)$$

because $h_{\mathcal{A}'}(t) = \{ h_{\mathcal{A}}(t) \}$. 
Equivalence of non-deterministic and deterministic $\Sigma$-automata

For the other direction, given $\mathcal{R} = \langle R, \rho \rangle$ such that $U = bh_\mathcal{R}(R_F)$, we construct the “subset automaton” $\mathcal{A} = \langle 2^R, \hat{\rho} \rangle$.

A simple inductive proof shows that $\mathcal{R}$ and $\mathcal{A}$ have the same output function $h_\mathcal{A}(t) = h_\mathcal{R}(t)$.

The final states of $\mathcal{A}$ are chosen as follows:

$$A_F = \{ u \mid u \subseteq R \text{ and } u \cap R_F \neq \emptyset \}$$

Therefore, $bh_\mathcal{A}(A_F) = bh_\mathcal{R}(R_F)$. 
Closure of the recognizable sets under the Boolean operations

If $U$ and $V$ are recognizable subsets of $T_{\Sigma}$, then $U \cap V$ and $T_{\Sigma} - U$ are recognizable.

**Proof.** Let $A$ and $B$ be $\Sigma$-automata such that $bh_A(A_F) = U$ and $bh_B(B_F) = V$, then:

(i) $bh_A(A - A_F) = T_{\Sigma} - U$

(ii) $bh_{A \times B}(A_F \times B_F) = U \cap V$
Closure under projection

**Definition:**
Let $\langle \Sigma, \sigma \rangle$ and $\langle \Omega, \omega \rangle$ be two species. A map $\pi: \Sigma \rightarrow \Omega$ satisfies the condition $\sigma(f) = \omega \pi(f)$ for all $f \in \Sigma$. The extended map $\bar{\pi}: T_{\Sigma} \rightarrow T_{\Omega}$ is the natural one.

Any mapping obtained from one species is called a *projection*. 
Closure under projection

**Theorem:** If $U$ is a recognizable subset of $T_\Sigma$, and $\bar{\pi}$ is a projection of $T_\Sigma$ into $T_\Omega$, then $\bar{\pi}(U)$ is a recognizable subset of $T_\Omega$. 
Closure under projection

Proof:
Let $\mathcal{A} = \langle A, \alpha \rangle$ be a $\Sigma$-automaton with $bh_\mathcal{A}(A_F) = U$.

We construct the non-deterministic $\Omega$-automaton $\mathcal{R} = \langle A, \rho \rangle$ with transition relations:

(i) For $\lambda \in \Omega_0$, $\rho_\lambda = \{ \alpha_\delta | \delta \in \Sigma_0 \text{ and } \pi(\delta) = \lambda \}$

(ii) For $\lambda \in \Omega_n$ and $a_1, ..., a_n, \in A$, $\rho_f(a_1, ..., a_n, a) \leftrightarrow \alpha_g(a_1, ..., a_n) = a$ for some $g$ with $\pi(g) = f$
Closure under projection

To prove the theorem we depend on the following two properties:

(I) For all $t \in T_{\Sigma}$, $h_{\mathcal{A}}(t) \in h_{\mathcal{R}}(\pi(t))$;

(II) For all $t' \in T_{\Omega}$, $a \in h_{\mathcal{R}}(t') \rightarrow h_{\mathcal{A}}(t) = a$ for some $t$ with $\pi(t) = t'$.

The proofs if (I) and (II) are both inductive and quite similar; we will prove only (II).
Closure under projection

(II) proof:
For \( t' = \lambda \in \Omega_0 \), if \( a \in \rho(t') = \rho_\lambda \) then by (i),
\[
a = \alpha_\delta \text{ for some } \delta \text{ with } \pi(\delta) = \lambda
\]
Thus, \( t = \delta \in \Sigma_0 \); \( h_\mathcal{A}(t) = \alpha_\delta \) and \( \pi(t) = \lambda \).

Now let \( f \) be an \( n \)-place function symbol of \( \Omega \) and assume (II) is true for terms \( t'_1, ..., t'_n \).
We are looking at the term \( t' = ft'_1 ... t'_n \).
If \( a \in h_\mathcal{R}(t') \), then by the definition of \( h_\mathcal{R} \), there exists \( a_1, ..., a_n \) with \( a_i \in h_\mathcal{R}(t'_i) \) such that \( \rho_f(a_1, ..., a_n, a) \).
This means (by (ii)) that there is a function symbol of \( g \in \Sigma \) with \( \pi(g) = f \) and \( \alpha_g(a_1, ..., a_n) = a \).

(i) For \( \lambda \in \Omega_0 \), \( \rho_\lambda = \{ \alpha_\delta | \delta \in \Sigma_0 \text{ and } \pi(\delta) = \lambda \} \)
(ii) For \( \lambda \in \Omega_n \) and \( a_1, ..., a_n, \in A \),
\[
\rho_f(a_1, ..., a_n, a) \leftrightarrow \alpha_g(a_1, ..., a_n) = a \text{ for some } g \text{ with } \pi(g) = f \]
Closure under projection

Now applying the induction hypothesis to the $t_i'$ we know that there are $\Sigma$-terms $t_1, \ldots, t_n$ with $\pi(t_i) = t_i'$ and $h_{\mathcal{A}}(t_i) = a_i$. Thus $t$ is taken to be $gt_1 \ldots t_n$ and we have $\bar{\pi}(t) = t'$ and

$$h_{\mathcal{A}}(t) = \alpha_g(h_{\mathcal{A}}(t_1), \ldots, h_{\mathcal{A}}(t_n)) = a.$$ 

**end of (II) proof.**

From (I) we find that if $t \in bh_{\mathcal{A}}(A_F)$ then $h_{\mathcal{A}}(t) \in h_{\mathcal{R}}(\pi(t))$ and thus $h_{\mathcal{R}}(\pi(t)) \cap A_F \neq \emptyset$ and $\pi(t) \in bh_{\mathcal{R}}(A_F)$, thus proving:

$$\bar{\pi}[bh_{\mathcal{A}}(A_F)] \subseteq bh_{\mathcal{R}}(R_F)$$

The converse inclusion comes from (II).
Closure under inverse projection

**Theorem:** If $U \subseteq T_{\Sigma}$ is recognizable, and $\pi: \Omega \to \Sigma$ is a projection, then $\bar{\pi}^{-1}(U)$ is a recognizable subset of $T_{\Omega}$.

**Proof:** Let $\mathcal{A} = \langle A, \alpha \rangle$ be a $\Sigma$-automaton such that $b h_\mathcal{A}(A_F) = U$. An $\Omega$-automaton $\mathcal{B} = \langle A, \beta \rangle$ is constructed with $\beta_f = \alpha_{\pi(f)}$ for all symbols $f \in \Sigma$.

By induction we will prove that

(1) $h_\mathcal{B}(t) = h_\mathcal{A}(\pi(t))$. 


Closure under inverse projection

- For $t = \lambda \in \Sigma_0$ this is true by the definition of $\beta$: $h_B(\lambda) = \beta_\lambda = \alpha_{\pi(\lambda)} = h_A(\pi(\lambda))$.

- Assuming that (I) is true for terms $t_1, \ldots, t_n$ we consider $t = ft_1 \ldots t_n$. Then, $h_B(t) = \beta_f(h_B(t_1), \ldots, h_B(t_n))$
  $= \alpha_{\pi(f)}(h_A(\pi(t_1)), \ldots, h_A(\pi(t_n)))$
  $= h_A(\pi(t))$.

$(I) \ h_B(t) = h_A(\pi(t))$. 
Closure under inverse projection

\[ t \in \bar{\pi}^{-1}(U) \iff \bar{\pi}(t) \in U \iff h_{\mathcal{A}}(\bar{\pi}(t)) \in A_F \iff h_{\mathcal{B}}(t) \in A_F \iff t \in bh_{\mathcal{B}}(A_F) \]

So we constructed an automaton \( \mathcal{B} \) that recognizes \( \bar{\pi}^{-1}(U) \).
The replacement lemma

If $\mathcal{A}$ is a $\Sigma$-automaton and $t_1, t_2, t, t'$ are $\Sigma$-terms such that $t'$ is obtained from $t$ by replacing an occurrence of $t_2$ by $t_1$, and if $h_{\mathcal{A}}(t_1) = h_{\mathcal{A}}(t_2)$, then $h_{\mathcal{A}}(t) = h_{\mathcal{A}}(t')$.

**Proof:** by induction on the construction of terms:
- if $t = \lambda$, then $t_2 = t$ and $t' = t_1$;

$h_{\mathcal{A}}(t) = h_{\mathcal{A}}(t_2) = h_{\mathcal{A}}(t_1) = h_{\mathcal{A}}(t')$. 
The replacement lemma

For $t = f t'_1 \ldots t'_n$:

- If the replacement occurs in one of the $t'_i$'s, then the induction hypothesis yields $h_A(t) = h_A(t')$.

- If $t_2 = t'_j$ - for convenience, say $t'_1$, then $t' = f t'_1 t'_2 \ldots t'_n$, and since $h_A(t'_1) = h_A(t')$, it is clear that $h_A(t) = h_A(t'_1)$. 

Corollary

Given non-overlapping occurrences of subterms \( t_1, \ldots, t_s \) of a term \( t \), if \( t' \) results from replacing \( t_i \) by \( t'_i \) and if

\[
  h_A(t_i) = h_A(t'_i) \quad (i = 1, \ldots, s),
\]

Then, \( h_A(t) = h_A(t') \)
The emptiness problem

Definition:

**Depth** of a term is defined recursively by:

(i) \( d(\lambda) = 1 \) for \( \lambda \in \Sigma_0 \)

(ii) \( d(f(t_1, \ldots, t_n)) = d(ft_1 \ldots t_n) = \max_i \{d(t_i)\} + 1 \)
The emptiness problem

**Notation:** $t \succcurlyeq t'$ means “$t$ has $t'$ as a subterm”

**Definition of subterm:**

(i) For $t = \lambda \in \Sigma_0$, $t \succcurlyeq t'$ iff $t' = \lambda$

(ii) For $f \in \Sigma_n$, $t_1, \ldots, t_n \in T_\Sigma$ and $t = ft_1 \ldots t_n$, $t \succcurlyeq t'$ iff $ft' = t$ or $t_i \succcurlyeq t'$ for some $i$, $1 \leq i \leq n$. 
The emptiness problem

**Theorem**: Let $\mathcal{A}$ be a $\Sigma$-automaton with $n$ states. If there exists a term $t$ such that $h_{\mathcal{A}}(t) = a$, then there exists a term $t'$ such that $d(t') \leq n$ and $h_{\mathcal{A}}(t') = a$.

**Proof**: 

if $d(t) \leq n$, then we’re done.

if $d(t) > n$, then there’s at least one sequence of subterms:

$$t = t_0 > t_1 > \cdots > t_{d(t)} \in \Sigma_0.$$
The emptiness problem

The corresponding sequence of states $a_i = h_A(t_i)$ must contain a repetition, say $a_i = a_j$ for $i < j$.

By the replacement lemma: if $t_j$ replaces $t_i$ in $t$ to obtain $t'$, then $h_A(t) = h_A(t')$.

If $d(t')$ is still greater than $n$, the process can be repeated until a reduced term is obtained which has a depth $\leq n$, and produces the output state $a$. 
The emptiness problem

- In order to determine whether or not for a $\Sigma$-automaton $\mathcal{A}$, $\text{bh}_\mathcal{A}(A_F) = \emptyset$, we test all terms $t$ with $d(t) \leq n$, where $n$ is the number of states of $\mathcal{A}$. If any term has an output state in $A_F$, $\text{bh}_\mathcal{A}(A_F) \neq \emptyset$, otherwise $\text{bh}_\mathcal{A}(A_F) = \emptyset$. 
Kleene theory – Regular language

- In conventional automata, we defined the set of regular languages \( R \) over \( \Sigma^* \) as:
  - all finite subsets of \( \Sigma^* \)
  - Closure over union, concatenation and Kleene star:

\[
A, B \in R \implies A \cup B \in R, A \cdot B \in R, A^* \in R
\]

- No other language is regular
Generalized Regularity

- To expand the definition of regularity to the generalized automaton, we define two new operations for any $\lambda \in \Sigma_0$:
  - Product $U \cdot \lambda V$, is the set of words obtained by replacing every occurrence of $\lambda$ in a word from $U$ with a word from $V$
  - Closure $U^\lambda$
Product examples

- \( U \cdot \lambda \{ \lambda \} = U \)

- \( U \cdot \lambda \phi \) is the set of words from \( U \) which do not involve \( \lambda \)

- If \( U_1 \subseteq U_2 \) and \( V_1 \subseteq V_2 \) then \( U_1 \cdot \lambda V_1 \subseteq U_2 \cdot V_2 \)
Closure

- The closure is defined as it is in conventional theory:
  \[ X^0 = \{ \lambda \} \]

- \[ X^{n+1} = X^n \cup U \cdot \lambda X^n \]

- \[ U^\lambda = \bigcup_{n=0}^{\infty} X^n \]
Closure - example

- Let $A_{\Sigma} = \{ f \lambda_1 \lambda_2 ... \lambda_n \mid \lambda_i \in \Sigma_0, f \in \Sigma_n \}$
- $A_{\Sigma}$ is the set of atomic terms
- If $\Sigma_0 = \{ \lambda \}$ then $A_{\Sigma}^\lambda = T_\Sigma$
- If $\Sigma_0 = \{ \lambda, \delta \}$ then $(A_{\Sigma}^\lambda)^\delta = (A_{\Sigma}^\delta)^\lambda = T_\Sigma$
Closure - example

- $\Sigma = \{g, \lambda, \delta\}, \Sigma_0 = \{\lambda, \delta\}, \Sigma_1 = \{g\}$
- $\{g \lambda\}^\delta = \{\delta, g \lambda\}$
- $\{g \lambda\}^\lambda = \{\lambda, g \lambda, gg \lambda, \ldots\}$
- $\left(\{g \lambda\}^\delta\right)^\lambda = \{\delta, g \lambda\}^\lambda = T_\Sigma$
- $\left(\{g \lambda\}^\lambda\right)^\delta = \{\delta\} \cup \{g \lambda\}^\lambda \neq T_\Sigma$
Regularity – first attempt

- Same as conventional automata theory
- All finite subsets of $T_\Sigma$ are $\Sigma$–regular
- Closure over union, concatenation and Kleene star:

$$A, B \text{ are } \Sigma \text{–regular, } \sigma \in \Sigma_0 \Rightarrow$$

$$A \cup B, A \cdot \lambda B, A^\lambda \text{ are } \Sigma \text{–regular}$$

- No other language is $\Sigma$–regular
Regularity – first attempt

- However, the following language \( V \) is recognizable but not \( \Sigma - regular \)

- \( \Sigma = \{ h, \lambda \}, \Sigma_2 = \{ h \}, \Sigma_0 = \{ \lambda \} \)

- Defined recursively:
  - \( \lambda \in V \)
  - \( t \in V \Rightarrow h\lambda t \in V \)
Regularity — second attempt

- However, given $\Sigma' = \{h, \lambda, \delta\}$ where $\Sigma_2 = \{h\}, \Sigma_0 = \{\lambda, \delta\}, V$ is $\Sigma'$–regular as:

  $$V = \{h\delta\lambda\}^{\lambda} \cdot \delta \{\lambda\}$$

- By adding a literal, we “made” the language regular
Regularity

\[ U \subseteq T_\Sigma \text{ is regular if there exists a species } \{\Sigma', \sigma'\} \text{ such that } \Sigma_n = \Sigma'_n \text{ for } n \geq 1 \text{ and } U \text{ is } \Sigma'\text{'-regular} \]
Regularity and Recognizability

- A set $U \in T_\Sigma$ is recognizable if and only if it is regular.

- The proof will be divided into two main parts:
  - The analysis theorem shows each recognizable set is regular.
  - The synthesis theorem shows each regular set is recognizable.
The analysis theorem - proof

- Let $\mathcal{A}$ be an automaton recognizing subset $U \subseteq T_\Sigma$ with states $A = \{a_1, \ldots, a_m\}$ and final states $A_F$.

- We construct a new symbol set:  
  $$\Sigma' = (\Sigma - \Sigma_0) \cup \{\lambda_1, \ldots, \lambda_m\}$$ 
  and an automaton $\mathcal{A}'$ which is exactly like $\mathcal{A}$ except that $a'_\lambda_i = a_i$ for $i = 1, \ldots, m$. 
The analysis theorem - proof

- To analyze $A'$, we will define a sequence of non-deterministic automaton

$A^1, ..., A^m = A'$

- $A^k = \langle A, \alpha^k \rangle$ where for $f \in \Sigma_n \alpha_f^k (a_1, ..., a_n)$ is defined if $\alpha_f (a_1, ..., a_n) \in \{a_1, ..., a_k\}$ and if defined, then $\alpha_f^k (a_1, ..., a_n) = \alpha_f (a_1, ..., a_n)$

- For any $k$ and any individual symbol $\lambda_i$

$\alpha_{\lambda_i}^k = a_i$
The analysis theorem - proof

- $A^k$ is the automaton with only the edges leading to $\{a_1, \ldots, a_k\}$

- $T_k^i = bh_{A^k}(\{a_i\}) = \{t \mid h_{A^k}(t) = a_i\}$

- We prove by induction on $k$ that $T_k^i$ is regular for $i = 1, \ldots, m$
The analysis theorem - proof

- For $k = 1$ and $i \neq 1$, $T_k^i = \{ \lambda_i \}$, which is clearly regular.

$$U_k = \left\{ f \lambda_{i_1} \lambda_{i_2} \ldots \lambda_{i_n} \mid f \in \Sigma_n \text{ and } \alpha_f (a_{i_1}, a_{i_2}, \ldots, a_{i_m}) = \alpha_k \right\}$$

- For $i = 1$, we claim that $T_1^1 = U_1^{\lambda_1}$.
The analysis theorem - proof

- $T_1^1 \subseteq U_1^{\lambda_1}$ is proved by induction on the construction of terms
- $U_1^{\lambda_1} \subseteq T_1^1$ is proved by induction on the sequence used in defining $U_1^{\lambda_1}$
The analysis theorem - proof

- The major inductive hypothesis is that $T_i^k$ is regular for $i = 1, \ldots, m$, and we want to prove that $T_i^{k+1}$ is regular.
- Of course, for $i > k + 1$ $T_i^k = \{\lambda_i\}$ which is regular
The analysis theorem - proof

- \( V = \left( \ldots \left( \left( U_{k+1} \cdot \lambda_1 T_1^K \right) \cdot \lambda_2 T_2^K \right) \ldots \right) \cdot \lambda_k T_k^K \)

- \( W = V^{\lambda_{k+1}} \)

- We want to prove that:

\[
T_i^{k+1} = T_i^k \cdot \lambda_{k+1} \quad W, \quad \text{for} \quad i = 1, \ldots, k + 1
\]
The analysis theorem - proof

- for \( t = f \lambda_{i_1} \ldots \lambda_{i_n} \in U_{k+1}, h_{A^{k+1}} (t) = a_{k+1} \)

- \( V = \left( \ldots \left( \left( U_{k+1} \cdot \lambda_1 T_1^k \right) \cdot \lambda_2 T_2^k \right) \ldots \right) \cdot \lambda_k T_k^k \)

- Therefore, by the replacement lemma we see that every term in \( V \) will also produce the output state \( a_{k+1} \)

- For the same reason \( t \in W \Rightarrow h_{A^{k+1}} (t) = a_{k+1} \)
The analysis theorem - proof

- Finally, if \( t \in T_i^k \cdot \lambda_{k+1} W \) it again follows from the replacement lemma that \( h_{A^k+1}(t) = i \).
- Thus, we have \( T_i^{k+1} \subseteq T_i^k \cdot \lambda_{k+1} W \).
- The other direction is harder. We will tackle it using induction over \( n \).
The analysis theorem - proof

- If $h_{A^{k+1}}(t) = a_i$ and there are exactly $n$ occurrences of non-initial transitions to $\alpha_{k+1}$ in the computation of $h_{A^{k+1}}(t)$, then

$$t \in T_i^k \cdot \lambda_{i+1} X^n$$

where

$$X^0 = \{ \lambda_{i+1} \}$$

$$X^{n+1} = X^n \cup V \cdot \lambda_{k+1} X^n$$
The analysis theorem - proof

- If there are no non initial transitions to $a_{k+1}$ then $t \in T_i^k = T_i^k \cdot \lambda_{k+1} \{\lambda_{k+1}\}$

- Assuming the claim is true for $n$, consider a term $t$ such that $h_{A^{k+1}}(t) = a_i$ and the computation of $h_{A^{k+1}}(t)$ has exactly $n+1$ non-initial transitions to $a_{k+1}$
The analysis theorem - proof

- Under this situation, there exist terms $t_1, t_2$ such that:

1. $h_{A^{k+1}}(t_1) = a_i$
2. $h_{A^{k+1}}(t_2) = a_{k+1}$
3. $t$ results from replacing one occurrence of $\lambda_{k+1}$ in $t_1$ with $t_2$
The analysis theorem - proof

- From (2) it follows that $t_2 = ft_1...t_n$

and for the appropriate choices of $\lambda_{ij}$

$f \lambda_{i_1} ... \lambda_{i_n} \in U_{k+1}$

- There can be no non-initial transition to $a_{k+1}$ in the computation of $h_{A^{k+1}}(t_i)$

and thus $t_j \in T_{ij}^k$, hence $t_2 \in V$
The analysis theorem - proof

- By (1) and the induction hypothesis
  \[ t_1 \in T_i^k \cdot \lambda_{k+1} X^n \]

- Combined with (3), we get
  \[ t \in \left( T_i^k \cdot \lambda_{k+1} X^n \right) \cdot \lambda_{k+1} V \]

- It is easily verified that
  \[ \left( T_i^k \cdot \lambda_{k+1} X^n \right) \cdot \lambda_{l+1} V \subseteq T_i^k \cdot \lambda_{k+1} X^{n+1} \]
The analysis theorem - proof

- Thus we have completed the inductions.
- Since \( \bigcup_n T_i^k \cdot \lambda_{k+1} X^n = T_i^k \cdot \lambda_{k+1} W \)
  and we have shown that \( T_i^{k+1} \subseteq \bigcup_n T_i^k \cdot \lambda_{k+1} X^n \)
  we get \( T_i^{k+1} \subseteq T_i^k \cdot \lambda_{k+1} W \)
- Thus \( T_i^{k+1} = T_i^k \cdot \lambda_{k+1} W \)
The analysis theorem - proof

- This completes the major inductive proof that $T_i^k$ is regular
- $bh_{\mathcal{A}'}(A_F) = \bigcup_{a_i \in A_F} T_i^k$
- Thus, $\mathcal{A}'$ is regular
- But we want to prove that $\mathcal{A}$ is regular
The analysis theorem - proof

- \( \Lambda_j = \alpha^{-1}(a_j) \)

- \( bh_{\mathcal{A}}(A_F) = bh_{\mathcal{A}'}(A_F) \cdot \lambda_1 \Lambda_1 \cdot \lambda_2 \Lambda_2 \cdots \lambda_2 \Lambda_m \)

- Thus, \( bh_{\mathcal{A}}(A_F) \) is regular

- And thus, the analysis theorem holds
The synthesis theorem

- We want to prove that any regular set $U \subseteq T^\Sigma$ is recognizable.
- We will divide this proof to 4 parts, based on the definition of the set of regular languages.
The synthesis theorem - proof

- All finite subsets of $T_{\Sigma}$ are recognizable: simple proof by constructing automata to recognize singleton sets, and with closure over union we get the claim.

- Closure of the recognizable sets under union – we already proved
The synthesis theorem - proof

• Closure of the recognizable sets under \( \cdot_\delta \)

• If \( U, V \subseteq \Sigma^*_T \) is recognizable, then \( U \cdot_\delta V \) is recognizable

• Let \( \mathcal{A} \) and \( \mathcal{B} \) be automatons with

\[
\text{bh}_{\mathcal{A}} \left( A_f \right) = U, \text{bh}_{\mathcal{B}} \left( B_f \right) = V
\]
The synthesis theorem - proof

- \( R = \langle A \cup B, \rho \rangle \) where:

- for \( \lambda \in \Sigma_0 \):

\[
\rho_\lambda = \begin{cases} 
\{\alpha_\delta, \beta_\lambda, \alpha_\lambda\} & \lambda \neq \delta \wedge \beta_\lambda \in B_F \\
\{\alpha_\lambda, \beta_\lambda\} & \lambda \neq \delta \wedge \beta_\lambda \notin B_F \\
\{\alpha_\delta, \beta_\delta\} & \lambda = \delta \wedge b_\delta \in B_F \\
\{\beta_\delta\} & \lambda = \delta \wedge \beta_\lambda \notin B_F 
\end{cases}
\]
The synthesis theorem - proof

- $R = \langle A \cup B, \rho \rangle$ where:

- for $f \in \Sigma_n, a_i \in A, b_i \in B, X \in R$

  $\rho_f (a_1, ..., a_n, X) \iff \alpha_f (a_1, ..., a_n) = X$

  or

  $\rho_f (b_1, ..., b_n, X) \iff \beta_f (b_1, ..., b_n) \in B_F \land X = \alpha_\delta$
The synthesis theorem - proof

- we claim that \( \text{bh}_R \left( A_F \right) = U \cdot V \)

- This assertion is a consequence of two propositions:

1. \( a \in h_R \left( t \right) \iff \exists t' \left( t \in \{ t' \} \cdot V \land h_A \left( t' \right) = a \right) \)

2. \( b \in h_R \left( t \right) \iff h_B \left( t \right) = b \)
The synthesis theorem - proof

- If \( t \in bh_R (A_F) \) then \( h_R (t) \cap A_F \neq \emptyset \)
- Thus there is \( a \in h_R (t) \) such that \( a \in A_F \)
- According to (1), there exists \( t' \in U \) such that \( t \in \{ t' \} \cdot \delta V \), thus \( bh_R (A_F) \subseteq U \cdot \delta V \)
- The converse \( U \cdot \delta V \subseteq bh_R (A_F) \) also follows from (1)
The synthesis theorem - proof

- We only need proposition (2) for the recursive proof of (1).

- For $\lambda \in \Sigma_0$, consider the case $t = \lambda \neq \delta$.

- If $a \in h_R(t)$ then either $a = \alpha_\lambda$ or $a = \alpha_\delta$
  
  and $\beta_\lambda \in B_F$
The synthesis theorem - proof

- In the first case, take $t' = \lambda$.
- In the second case, take $t' = \delta$.
- If $t = \delta$ then $a \in h_{R}(t)$ if and only if $a = \alpha_{\delta}$ and $\beta_{\delta} \in B_{F}$. In this case, take $t' = \delta$.
- Proof of second direction done in the same manner
The synthesis theorem - proof

• (2) Is easily verified since

\[ b \in h_R(\lambda) \iff b = \beta_\lambda \text{ and } h_B(\lambda) = \beta_\lambda \]

• Next, the inductive part. Assume \( f \in \Sigma_n \) and the induction claim holds for \( t_1, \ldots, t_n \).

• \( b \in h_R(ft_1 \ldots t_n) \) iff there exists \( b_i \in h_R(t_i) \) such that \( \beta_f(b_1, \ldots, b_n) = b \), and from the induction claim \( h_B(ft_1 \ldots t_n) = b \).
The synthesis theorem - proof

- Let us say \( t = f t_1 \ldots t_n \)

- If \( a \in h_R (t) \) there are two cases to consider:
  - Either there exists \( a_i \in h_R (t_i) \) with
    \[ \alpha_f (a_1, \ldots, a_n) = a \]
  - Or there exists \( b_i \in h_R (t_i) \) such that
    \[ \beta_f (b_1, \ldots, b_n) \in B_F \text{ and } a = \alpha_\delta \]
The synthesis theorem - proof

• For the first case, the inductive hypothesis (1) assures the existence of $t'_i$ with

$$t_i \in t'_i \cdot \delta V$$

• Thus we can take $t' = ft'_1 ... t'_n$ and then

$$t \in t' \cdot \delta V \text{ and }$$

$$h_{\mathcal{A}} (t') = \alpha_f (h_{\mathcal{A}} (t'_1),...,h_{\mathcal{A}} (t'_n)) =$$

$$= \alpha_f (a_1,...,a_n) = a$$
The synthesis theorem - proof

- For the latter case, the inductive hypothesis (2) gives us $h_B(t_i) = b_i$ and therefore $h_B(t) \in B_F, t \in V$
- So we can take $t' = \delta$ and then $t \in \{\delta\} \cdot \delta V$
  and $h_R(\delta) = \alpha_\delta = a$
- We have established one direction of (1)
  The other is similar, based on the same separation to cases.
The synthesis theorem - proof

- Closure of the recognizable sets under $\delta$
- If $U \subseteq T_\Sigma$ is recognizable then $U^\delta$ is recognizable
- Let $A$ be an automaton such that $bh_A(A_F) = U$
The synthesis theorem - proof

- $\mathcal{R} = \langle A, \rho \rangle$ where:

- For $\lambda \in \Sigma_0 : \rho_\lambda = \{ \alpha_\lambda \}$

- For $f \in \Sigma_n, a_i, x \in A :$

  \[ \rho_f (a_1, ..., a_n) \iff \alpha_f (a_1, ..., a_n) = x \vee \left( \alpha_f (a_1, ..., a_n) \in A_F \land x = \alpha_\delta \right) \]

- The proof that $bh_{\mathcal{R}} (A_F) = U^\delta$ is similar to the proof of closure over concatenation.
The synthesis theorem - proof

- We proved that all finite languages are recognizable, and that recognizable languages are closed over union, concatenation and $\delta$-closure. Thus, every regular set is recognizable.
Regularity and Recognizability

- From the analysis theorem and the synthesis theorem we get that regularity and recognizability are equivalent.
Multiple Successor Arithmetics

- \( A_k = \{1, \ldots, k\} \)
- \( N_k \) - set of strings over \( A_k \)
- \( \Lambda \) - empty string
- \( r_1, \ldots, r_k \) - successor function where:
  \[
  r_\sigma(w) = w\sigma, \Lambda\sigma = \sigma
  \]
Weak Monadic Second-order Theories

The language of sentences $\ell_k$ is:

- Individual variables $x, y, z, \ldots$, ranging over $N_k$
- Set variables $\alpha, \beta, \gamma, \ldots$, ranging over finite subsets of $N_k$
- Constants $=, \in$ with their usual interpretation
Weak Monadic Second-order Theories

- Constant binary predicate symbol $R_\sigma$
  interpreted as the graphs of the $k$-successor functions

- Propositional connectives, individual and set quantifiers, punctuation and parentheses as required for the applied language.
Weak Monadic Second-order Theories

- Atomic formulas:
  \[ x = y, x \in \alpha, R_\sigma (x, y) \]

- If \( F, G \) are formulas then the following are formulas:
  \[ F \land G, \neg F, \exists x F, \exists \alpha F \]

- A sentence is a formula with no free variables
Decidability of Weak Monadic Second-order Theories

- The weak monadic second-order logic of one successor is decidable.
- What about $k \geq 2$?
- From now on, we will assume $k = 2$. The generalization to general $k$ is simple.
Decidability of Weak Monadic Second-order Theories

The procedure will be as follows:

1. Find a language $\ell_2'$ equivalent to $\ell_2$ which has no individual variables or quantifiers.

2. Encode n-tuples of finite subsets of $\mathbb{N}_2$ where $\Sigma_0^n = \{ \lambda \}$ and $\Sigma_2^n = \{ 0, 1 \}^n$; $
\Sigma^n = \Sigma_0^n \cup \Sigma_2^n$

3. Show that every definable relation of $\ell_2'$ is recognizable when interpreted under the encoding of (2)