B Infinite words

<u>Learned</u>: · Finite automata A, WMSO formulas φ, and their languages.

· Now $A \models \varphi$ defined by $\mathcal{L}(A) \subseteq \mathcal{L}(\varphi)$ makes sense:

"all words of A are allowed by φ"

⇒ <u>Model checking</u>

· A usually called <u>system</u>

· φ usually called <u>specification</u>

· <u>Check</u> if A is a <u>model</u> of φ (in the $\models$ sense)

<u>This chapter</u>: · Typically, finite words are not sufficient

↳ OS not meant to terminate

↳ $\Box \Diamond$ req ← where a request is issued

in every position / there is a later position

· New class of automata: <u>Büchi</u> automata (system) (next lectures)

· New logic: LTL (linear-time temporal logic) (<u>specification</u>) (afterwards)

· More complex <u>system</u> models: Büchi pushdown automata

↳ Büchi automata come without recursion. (end of B)

4. ω-regular languages and Büchi automata

<u>Goal</u>: Recognise sets of <u>infinite</u> words with finite automata

↳ What is an accepting run? Final state reachability fails!

↳ <u>Büchi condition</u>: visit final states infinitely often.

<u>Algorithmic problems</u>:

↳ Does automaton accept a word? (emptiness)

⇒ Model checking

↳ Do automata A and B accept the same language? (equivalence)

Key challenge: determinisation
Application: · Model checking MSO
↳ second-order quantifiers ranging over infinite sets (not much on this)
· LTL as syntactic fragment of MSO.

4.1 ω-regular languages

Let Σ an alphabet.

Notions:

ω-word = infinite sequence $a_0 a_1 a_2 \ldots$ with $a_i \in \Sigma$ f.a. $i \in \mathbb{N}$

Σ^ω = set of all ω-words over Σ

ω-language $L \subseteq \Sigma^\omega$ = set of ω-words

Let $w \in \Sigma^\omega$. Then $|w|_a \in \mathbb{N} \cup \{\omega\}$ = number of as in w.

Concatenation:

Impossible to concatenate $v, w \in \Sigma^\omega$.

But:

If $v \in \Sigma^*$ and $w \in \Sigma^\omega$ then $v.w \in \Sigma^\omega$.

Let $V \subseteq \Sigma^*$ and $W \subseteq \Sigma^\omega$. Then $V.W := \{v.w \mid v \in V, w \in W\} \subseteq \Sigma^\omega$

Let $v \in \Sigma^+$. Then

$$v^\omega := v.v.v.v\ldots$$

Let $L \subseteq \Sigma^*$ with $L \cap \Sigma^+ \neq \emptyset$. Then

$$L^\omega := \{v_0.v_1.v_2.v_3\ldots \mid v_i \in L \setminus \{\varepsilon\} \text{ f.a. } i \in \mathbb{N}\}.$$

Example:

Set of all words with
· infinitely many bs so that
· two bs are separated by even number of as
is

$$a^*.((aa)^*.b)^\omega$$

Define <u>ω-regular languages</u>

↳ Choose infinite iteration of regular languages

↳ "Correct definition" in following sense

⇒ Has natural corresponding automaton model

<u>Definition (ω-regularity)</u>

A language $L \subseteq \Sigma^\omega$ is called <u>ω-regular</u> if there are regular languages $V_0, \dots, V_{n-1}, W_0, \dots, W_{n-1} \subseteq \Sigma^*$ with $W_i \cap \Sigma^+ \neq \emptyset$ f.a. $0 \leq i \leq n-1$ so that

$$L = \bigcup_{i=0}^{n-1} V_i . W_i^\omega$$

<u>Example:</u>

Let $\Sigma = \{a, b, c\}$. Then

$$L = \{ w \in \Sigma^\omega \mid \underbrace{|w|_a = \omega}_{\text{if there are infinitely many as}} \Rightarrow \underbrace{|w|_b = \omega}_{\text{then there are infinitely many bs.}} \}$$

is ω-regular by

$$L = \underbrace{((a \cup c)^* . b)^\omega}_{\text{infinitely many bs}} \cup \underbrace{(a \cup b \cup c)^* . (b \cup c)^\omega}_{\text{finitely many as}}$$

This is the implication rewritten as

not $|w|_a = \omega$ or $|w|_b = \omega$

<u>Lemma:</u>

The class of ω-regular languages is closed under

- union
- concatenation from left with regular languages (use distributivity of $.$ over $\cup$)

For remaining closure properties: automata helpful.

4.2 Büchi automata

- Syntactically finite automata
- Acceptance condition changed

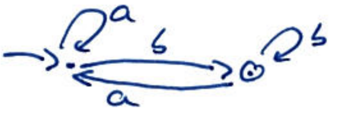
Definition (Büchi automaton):

- A non-deterministic Büchi automaton (NBA) over Σ is a tuple
$$A = (Q, q_0, \rightarrow, Q_F)$$
with the usual states Q, initial state $q_0 \in Q$, final states $Q_F \subseteq Q$, and transition relation $\rightarrow \subseteq Q \times \Sigma \times Q$.
- Run of A is an infinite sequence
$$r = q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} \dots$$
If $w = a_0 a_1 a_2 \ldots \in \Sigma^\omega$, say we have a run of A on w.
Write $q_0 \xrightarrow{w}$ if intermediary states exist but are unimportant (but there is a run of A on w)
- Let
$$\mathrm{Inf}(r) = \text{states that occur infinitely often in } r$$
Run r is accepting if
$$\mathrm{Inf}(r) \cap Q_F \neq \emptyset$$
- ω-language of A is
$$L(A) := \{ w \in \Sigma^\omega \mid \text{there is an accepting run } r \text{ of } A \text{ on } w \}.$$

Comment:

Acceptance = one final state visited infinitely often
= the set of final states visited infinitely often
($\Leftarrow$ finiteness of final states).

Example I:

Let $\Sigma = \{a, b\}$. Consider $L_1 = (a^* . b)^\omega$ // Infinitely many bs
Accepted by A_1 

(claim: $L(A_1) = L_1$.

Example 2:

Let $\Sigma = \{a, b\}$. Consider $L_2 = (a \cup b)^* a^\omega$ // finitely many bs

Then NBA

A_2 

satisfies $L(A_2) = L_2$.

- Note that $L(A_2) = \overline{L(A_1)} = \{a, b\}^\omega \setminus L(A_1)$
- Moreover, A_2 is non-deterministic while A_1 was deterministic.

Definition (Deterministic Büchi automata)

An NBA $A = (Q, q_0, \rightarrow, Q_F)$ over Σ is called deterministic or DBA if

for all $q \in Q$ and all $a \in \Sigma$ there is precisely one q' with $q \xrightarrow{a} q'$.

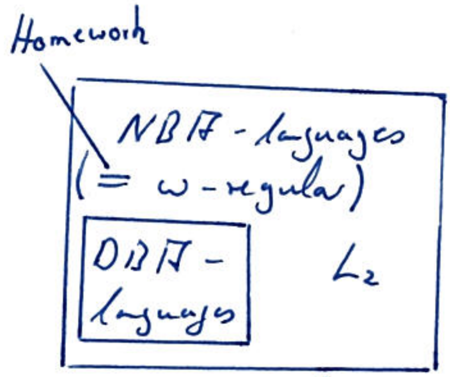
Not by accident that A_2 is NBA while A_1 was DBA.

$\Rightarrow$ $L(A_2)$ can not be recognised by DBA

$\Rightarrow$ In sharp contrast to NFA = DFA.

Theorem:

There are languages recognised by NBAs but not by DBAs.



(But we talked about Safra $\rightsquigarrow$ acceptance condition)

Proof:

Consider $L_2 = \{w \in \{a, b\}^\omega \mid |w|_b < \omega\}$

Towards a contradiction, assume L_2 was accepted by a DBA

$A = (Q, q_0, \rightarrow, Q_F)$, i.e., $L_2 = L(A)$.

Consider

$w_0 = b.a^\omega \in L_2$

$(L_2 = L(A))$
$\Rightarrow$ There is an accepting run r_0 of A on w_0:

$r_0 = q_0 \xrightarrow{b} q_1 \xrightarrow{a} q_2 \xrightarrow{a} \ldots$

Let $i_0 \in \mathbb{N}$ so that

after b and a^{i_0} arrive in final state

(this i_0 exists as the run is accepting).

Consider

$w_1 = b a^{i_0} b a^\omega \in L_2$

=> There is an accepting run r_1 of A on w_1

$$r_1 = q_0 \xrightarrow{b} q_1 \xrightarrow{a^{i_0}} q_{1+i_0} \xrightarrow{b} q_{1+i_0+1} \xrightarrow{a} \dots$$

Since A is deterministic, r_0 and r_1 coincide on first $i_0 + 2$ states (q_0 up to q_{1+i_0}).

Let $i_1 \in \mathbb{N}$ so that

after last b and a^{i_1} arrive in final state.

Consider

$w_2 = b a^{i_0} b a^{i_1} b a^\omega \in L_2$

...

Construction results in infinite word

$w = b.a^{i_0}.b.a^{i_1}.b a^{i_2}.b a^{i_3}\dots \in \{a,b\}^\omega$.

Since A is deterministic, there is a unique run r on w.

Claim: run r is <u>accepting</u>

=> Has a common prefix with all r_i on w_i.

=> Every new prefix $r_i \rightsquigarrow r_{i+1}$ yields another final state.

=> Run r visits final states infinitely often.

Thus, $w \in L(A)$, in contradiction to $w \notin L_2$ because of the infinitely many bs. $\square$

<u>Consequence:</u>

There are NBAs that cannot be determinised (into DBA).

Since $L_2 = (a \cup b)^* . a^\omega$, may assume that

$$\underbrace{\omega\text{-regular languages}}_{\text{expressions / closure}} = \underbrace{\text{NBA - recognisable}}_{\text{automata}}.$$

This in fact holds.