Literatur:

↳ Automatentheorie und Logik, A. Hofmann, M. Lange, Springer

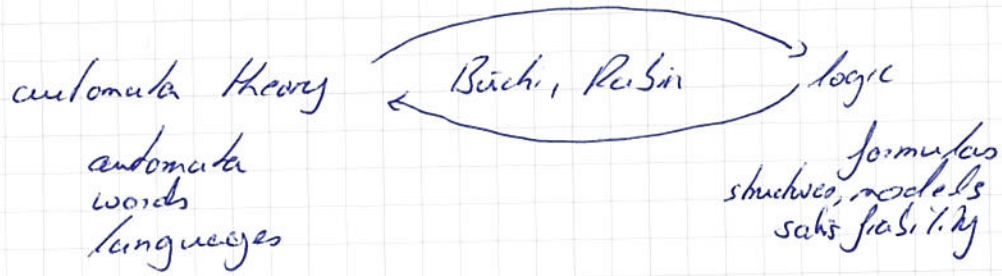
Automatentheorie und Logik Skript, E. Dost, Uni. Oldenburg

Languages, automata, and logic, Handbook of formal languages
W. Thomas

↳ Papers, Links online: http://concurrency.cs.uni-bluete/teaching.html

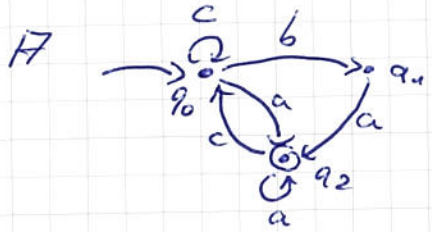
Topics of this course:

(H) Connection between automata theory and logic



What could such a connection look like?

Consider following finite automaton:



Language of A consists of the words where

(1) There is no b following an a.

(2) Every b is followed by an a.

(3) a is the last letter.

„⊆" Take a look „⊇" · By (3), run ends in q2 (if not stuck

· In q0 it doesn't get stuck

· By (2), doesn't get stuck in q1

· By (1), doesn't get stuck in q2.

Can we characterize $L(M)$ logically?

↳ possible?

↳ how to do this algorithmically?

Possible = Yes:

↳ Particular logic on words

↳ For the definition, understand words e.g. $a = : w$ as function from positions to letters

$w(0) = c$
$w(1) = b$
$w(2) = a$

$w : \{0,1,2\} \to \{a,b,c\}$

Shift in view to words

Use predicates

$Q_a(x)$ "at position x you find a"

$S(x,y)$ "position y follows x"

Consider

$$\varphi_M = \neg \exists x \exists y : (S(x,y) \wedge Q_a(x) \wedge Q_b(y))$$
$$\wedge \forall x : (Q_b(x) \to \exists y : (S(x,y) \wedge Q_a(y)))$$
$$\wedge \exists x : \neg \exists y : S(x,y) \wedge Q_a(x)$$

Claim:

Words that satisfy φ_M are precisely the words in $L(M)$.

More generally, language definable in logic, if there is a formula that characterizes the language (in this way).

Büchi - Elgot:

Language is regular iff MSO definable

(iff NFA recognisable
iff DFA recognisable
iff reg. exp. generated
iff left. - linear grammar
iff right - linear grammar

} I-G DP.

(B) More than words

Automata theory and logics can be studied for various objects and with different purposes:

Finite words:

Applications:

Verification of recursive programs

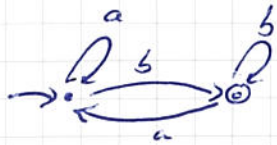
↳ Represent reachable states:

$(q, \perp) \to (q, \square), (q, \square), (q, \square), (q, \square)$

Stack content: $\Gamma^* \cdot \gamma^*$

Bourajjani.

Infinite words:

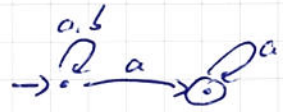
A 

"Every infinite run has to visit infinitely many final states."

$L(A) = (a^* \cdot b)^\omega$ "Words with infinitely many bs"

Complement: "Words with finitely many bs"

$\bar{A}$



Automaton $\bar{A}$ non-deterministic:

↳ By accident?

↳ No! There is no deterministic Büchi automaton A_1 with $L(A_1) = \overline{L(A)}$.

↳ Need that for NFAs there always is a corresponding DFA.

Applications:

Verification of reactive systems:

↳ Should not terminate but interact forever with environment

- $Sys \models prop$ Model-Checking

$prop = \Box \Diamond req \wedge \Box(req \Rightarrow \Diamond ack)$

always eventually

Sys — req req ack req ack — w

- Does the system terminate?

Finite trees:

Tree language = set of trees

What are the tree languages recognised by finite automata?

↳ How does an automaton work on trees?

=> Bottom up or top down

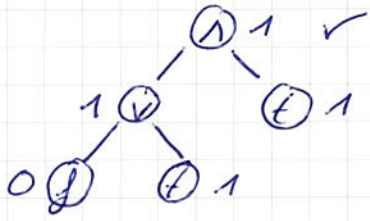
- Label tree from leaves to root by states (bottom up)
- Accept if root labelled by final state

$A = (\underbrace{\{0,1\}}_{\text{states}}, \delta, \underbrace{\{1\}}_{\text{final state}})$

$\delta_0() := \{0\}$, $\delta_1() := \{1\}$

$\delta_\vee(q_1, q_2) = \{q\}$ with $q = \max\{q_1, q_2\}$

$\delta_\wedge(q_1, q_2) := \{q\}$ with $q = \min\{q_1, q_2\}$

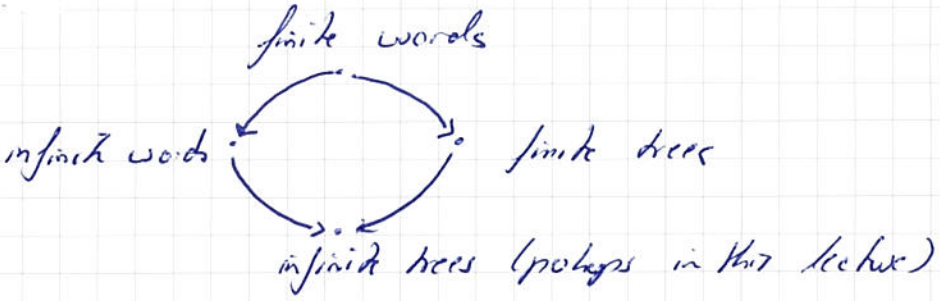


$L(A)$ = "all variable-free Boolean expressions that evaluate to true".

Applications:

Type checking in XML-documents
(XML-schema)

Overview:



Thus we see the objects that automata and logics may work on:

- ↳ partially ordered words (unfoldings)
- ↳ graphs.