ECE 641
Advanced Topics in Supervisory Control for Discrete Event Systems

Lecture 1

Associate Prof. Dr. Klaus Schmidt

Department of Mechatronics Engineering – Çankaya University

PhD Course in Electronic and Communication Engineering
Credits (3/0/3)
Course webpage: http://ece641.cankaya.edu.tr/

Content and Structure

Content

- Monolithic Diagnosability and Diagnoser
- Computational Complexity
- Decentralized Diagnosis
- Diagnosis and Abstractions
- Supervisory Control under Partial Observation
- Decentralized Supervisory Control
- Petri Nets

Structure

- 3 lecture hours: Wednesday 18:40 – 21:30
- Office hours: Monday 12:30 – 13:30

Grading and Literature

Grading

- Presentation of recent research paper (30%)
- 1 Midterm Exam (30%)
- 1 Final Exam (40%)

Literature

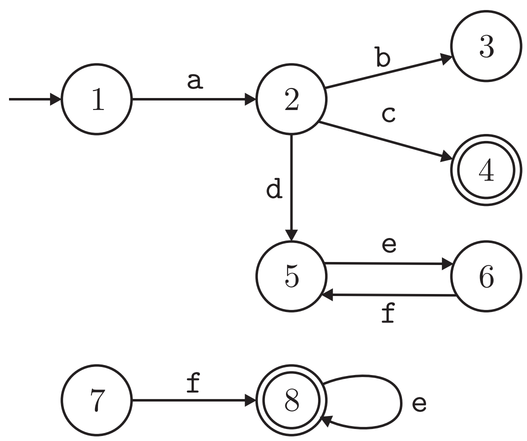
- Christos G. Cassandras and Stephane Lafortune: Introduction to Discrete Event Systems Springer, New York 2008 (ISBN 0-38-733332-0) (Main Textbook)
- Supervisory Control of Discrete Event Systems Using Petri Nets, Kluwer Academic Publishers (ISBN: 0-7923-8199-8), 1998.
- T.-S. Yoo, H. E. Garcia Diagnosis of behaviors of interest in partially-observed discrete-event systems System & Control Letters, 2008.
- W. Qiu and R. Kumar: Decentralized failure diagnosis of discrete event systems, IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 36, no. 2, pp. 384395, 2006.

Automata Modeling: Definition

Finite State Automaton

- Five-tuple: $G = (X, \Sigma, \delta, x_0, X_{\mathrm{m}})$
- Set of *states*: X
  $\rightarrow$ Circles in graph
- *Alphabet* of events: Σ
- *Transition relation*: $\delta : X \times \Sigma \rightarrow X$
  $\rightarrow$ Arcs labeled with events
- *Initial state*: $x_0 \in X$
  $\rightarrow$ State with incoming arrow
- Set of *marked states*: $X_{\mathrm{m}} \subseteq X$
  $\rightarrow$ States with double circle

Automata Graph



Automata Modeling: Semantics

Semantics for DES Modeling

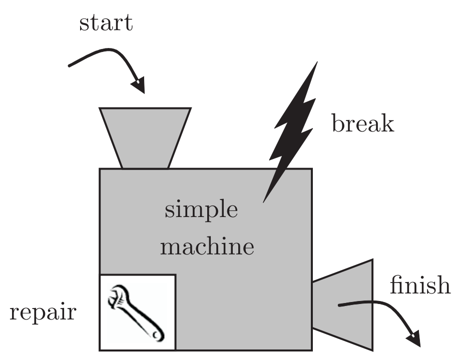
- State: Operational modes of a DES
  → System can spend time in each operational mode
- Transitions: Instantaneous change of the system state
  → No time is spend when a transition is executed
- Events: Labels of transitions
  → Transitions are executed when associated event occurs
- Initial state: State where the system operation starts
- Marked states: Distinguished (special) states that should be reached
  → It is generally desirable to reach marked states for example to complete a task

Introduction: Simple Machine Example

Verbal Description

- Machine can start processing if it is idle
- Machine finishes processing after some time
- Machine can break while processing
- Machine can be repaired after breakdown
- Machine can restart after repair

Simple Machine



DES Characterization

Gap 1

Introduction: Simple Machine Example

DES Characterization

Gap 2

Automata Modeling: Simple Machine Example

Automaton Model

Gap 3

Formal Languages: Definition

Strings

- Given an alphabet Σ, a string is a finite sequence of events from Σ
- The empty string is ϵ

Language

- The set of all possible strings over Σ is called Σ^* (Kleene Closure)
- Given an alphabet Σ, any subset of Σ^* is a formal language
  $\rightarrow$ Set of strings over Σ

Illustration

Gap 4

autonomous and non-autonomous systems

Formal Languages: Operations

Concatenation

- The concatenation st of two strings $s \in \Sigma^*$ and $t \in \Sigma^*$ is the string that is obtained when attaching t to the end of s

Prefix Closure

- For a string $s \in \Sigma^\star$, s' is a prefix of s if there is $t \in \Sigma^\star$ such that $s = s't$
- For a language $L \subseteq \Sigma^*$, the prefix-closure $\overline{L}$ contains all prefixes of strings of L

$$\overline{L} = \{s \in \Sigma^* | st \in L \text{ for some } t \in \Sigma^*\}$$

Illustration

Gap 5

Formal Languages: Automata Languages

Closed Language $L(G)$

- For $G = (X, \Sigma, \delta, x_0, X_\mathrm{m})$, $L(G)$ contains all event sequences that follow transitions starting from the initial state of G

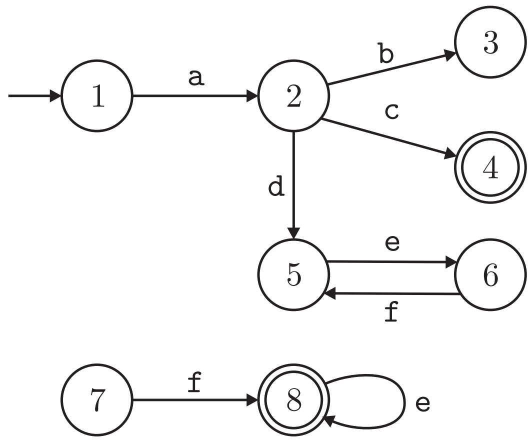
$$L(G) = \{s \in \Sigma^* | \delta(x_0, s) \text{ exists}\}$$

Marked Language $L_\mathrm{m}(G)$

- For $G = (X, \Sigma, \delta, x_0, X_\mathrm{m})$, $L_\mathrm{m}(G)$ contains all event sequences leading from the initial state to a marked state of G

$$L_\mathrm{m}(G) = \{s \in L(G) | \delta(x_0, s) \in X_\mathrm{m}\}$$

Automata Graph



Fact

$$L_\mathrm{m}(G) \subseteq L(G)$$

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Formal Languages: Automata Languages

Illustration

Gap 6

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Automaton Properties: Nonblocking

Accessible

- $G = (X, \Sigma, \delta, x_0, X_{\mathrm{m}})$ is accessible if every state in X is reachable from the initial state

$$\forall x \in X, \exists s \in \Sigma^* \text{ such that } \delta(x_0, s) = x$$

Coaccessible

- $G = (X, \Sigma, \delta, x_0, X_{\mathrm{m}})$ is coaccessible if from every state in X a marked state is reachable

$$\forall x \in X, \exists s \in \Sigma^* \text{ such that } \delta(x, s) \in X_{\mathrm{m}}$$

- Coaccessible automata are also called nonblocking

$$\overline{L_{\mathrm{m}}(G)} = L(G)$$

Trim

- $G = (X, \Sigma, \delta, x_0, X_{\mathrm{m}})$ is trim if it is accessible and coaccessible

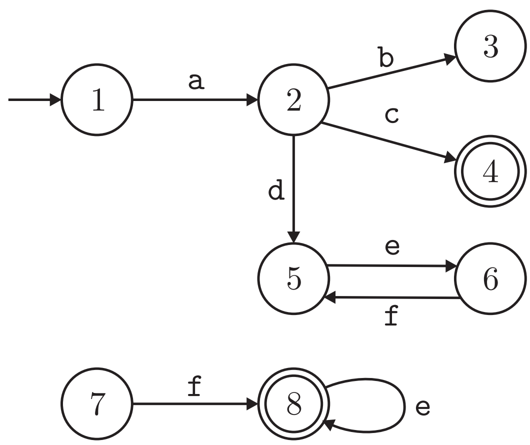
Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Automaton Properties: Example

Properties

Gap 7

Automata Graph



Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Regular Languages: Definition

Regular Language

- A language is regular if it is recognized by a finite state automaton

Example

Gap 8

Syllabus Reminder Automata Modeling Formal Languages Automaton Properties Regular Languages

Regular Languages: Regular Expression

Notation

- For string u: $u^\star = \{u\}^\star = \{\epsilon, u, uu, uuu, uuuu, \ldots\}$ (Kleene closure)
- For strings u, v: $uv = \{uv\}$ (Concatenation)
- For strings u, v: $u + v = \{u\} \cup \{v\}$ (Union)

Regular Expressions

1. $\emptyset$ is a regular expression denoting the empty set, ϵ is a regular expression denoting the set $\{\epsilon\}$, and σ is a regular expression denoting the set $\{\sigma\}$, for all $\sigma \in \Sigma$
2. If r and s are regular expressions, then rs, $r + s$, $r^\star$ are regular expressions.
3. There are no regular expressions other than those constructed by applying rules 1 and 2 above a finite number of times.

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Regular Languages: Example

Computation

Gap 9

Regular Languages: Pumping Lemma

Lemma (Pumping Lemma)

Let $L \subseteq \Sigma^\star$ be a regular language. Then there exists an integer $p = 1$ depending only on L such that every string $s \in L$ with $|s| \geq p$ can be written as $s = xyz$, satisfying the following conditions:

- $|y| \geq 1$
- $|xy| \leq p$
- *for all* $i \geq 0$, $xy^iz \in L$

Remarks

- $s \in L$ can be divided into three substrings
- y corresponds to a loop that can be repeated arbitrarily
- p is called the pumping length

Regular Languages: Example

Pumping Lemma

Gap 10

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University

Regular Languages: Example

Non-regular Language

Gap 11

Klaus Schmidt
Department of Electronic and Communication Engineering – Çankaya University