

Hesaplama Teorisi

2025--2026 Fall
Zeynep Altan

Bu ders hangi sorulara cevap verir?

- Bir hesaplama problemini çözebilmek için gerekli kaynaklar (**zaman, mekan, iletişim ve işlemci sayısı** vs.) problemin büyüklüğüne göre nasıl ölçeklenir?
- Eğer bir problem çözülemezse, belli bir zaman aralığında yaklaşık çözümlerini bulabilmek mümkün müdür?
- Gerçek olan rastgele sayıları, deterministik yollarla üretilen rastgele görünen sayılardan **ayırt edebilmek** mümkün müdür?
- Gerçek rastgeleliğe** gerek var mıdır?

Bu ders hangi sorulara cevap verir?

- Sadece gizli mesajların deęişimi deęil, modern elektronik ticaret için gerekli dięer pek çok görevde güvenli kriptografik protokoller oluşturulabilir mi?
- Daęıtılmış ajanlar (temsilciler/ agents), hatalar (faults) ve kötü niyetli müdahaleler (malicious interference) olması durumunda nasıl kararlar alınabilir?
- Bir beyindeki nöronlar, bir hücredeki proteinler veya bir pazaryerindeki toplu olarak alıcılar ve satıcılar ne tür hesaplamalar gerçekleştirebilirler?
- Kuantum bilgisayarların yetenekleri ve sınırları nelerdir?

Hesaplama Teorisinin Güncel Uygulama Alanları hangi teorileri kullanır

Automata

Formal Languages

Computability

Complexity

Logic

Güncel uygulama Alanları: LLM ve Güvenlik

□ Formal languages, automata, program analysis, verification, complexity

Örnekler:

- ❖ Prompt filtering and jailbreak detection using finite-state/regex pipelines before ML inference (e.g., OpenAI/enterprise safety filters)
- ❖ SMT/SAT ve soyut yorumlama (Marabou, ERAN, AlphaBeta-CROWN) kullanılarak sinir ağı özelliklerinin (sağlamlık, monotonluk gibi) formal doğrulaması (formal verification) yapılır.
- ❖ JSON/SQL/XML çıktısını sağlamak için düzgün diller/CFG'lerle sınırlandırılmış olarak kod çözülme; Outlines (Python kütüphanesi) ve LMQL (Language Model Query Language) ile otomatlara kısıtlamalar getirilir ve kod çözme yönlendirilir.

Jailbreaking and Mitigation of Vulnerabilities in Large Language Models

<https://www.researchgate.net/publication/384972446> Jailbreaking and Mitigation of Vulnerabilities in Large Language Models

Güncel Uygulama Alanları: Blockchain Consensus and Protocols

Dağıtık hesaplanabilirlik, olası olmayan sonuçlar (FLP), karmaşıklık
Distributed computability, impossibility results (FLP), complexity

Application: Protocols balance liveness and safety under network assumptions; proofs rely on formal computation

HotStuff: BFT Consensus in the Lens of Blockchain

<https://arxiv.org/pdf/1803.05069>

Thunderella: Blockchains with Optimistic Instant Confirmation

<https://eprint.iacr.org/2017/913.pdf>

Güncel Uygulama Alanları: Human–computer interaction and editors

Regular languages, PDAs, incremental parsing

Tree-sitter incremental parsers and DFA lexers drive real-time syntax highlighting, refactoring, and structural search in VS Code/Neovim.

<https://tomassetti.me/incremental-parsing-using-tree-sitter/>

Fuzzy search uses bit-parallel approximate matching (Myers' algorithm).

Diğer Uygulama Alanları

Text Search and Code Editors

Network Intrusion Detection

Compilers and programming Languages

Natural Language Processing (parsing)

Data Compression

Model Checking for Software/Hardware Verification

Route Planning and Maps

Hesaplama Teorisi (HT) ile ne yapılır?

HT'nin temel yapısından ve problemlerinin özünden ayrılmadan biyoloji, ekonomi, fizik, hesaplamalı biyoloji, makine öğrenmesi, hesaplamalı ekonomi, dağıtık ve paralel hesaplama, istatistik, sayısal hesaplama, oyun teorisi gibi birçok alanda uygulanmaktadır.

- P (polinom zamanlı) ile NP (belirsiz polinom zaman) problemi, *graf teorisi*, cebir ve geometrideki geleneksel problemlerle bağlantılı karmaşıklıklardır.
 - ❖ P karmaşıklık sınıfındaki problemler, polinom zamanında deterministik bir Turing makinesi tarafından etkin olarak çözülebilen karar problemleridir.
 - ❖ NP, belirsiz polinom zamanlı problemler, çözümleri polinom zamanında verimli bir şekilde doğrulanabilen hesaplamalı problemlerdir.

P Probleminin Doğrulaması

- ❑ Bir çözümün doğruluğu polinom zamanda doğrulanabilir ve önerilen çözümün önemli bir hesaplama çabası gerektirmeden doğru olduğundan emin olunabilir.
- ❑ Örneğin sıralama algoritmaları, arama algoritmaları ve efektif çözümleri bilinen problemler verilebilir.

NP Probleminin Doğrulanması

- ❑ Olası herhangi bir çözüm sağlandıktan sonra, bu problemin doğruluğu bir polinom zamanında belirsiz bir algoritma kullanılarak hızlı bir şekilde doğrulanabilir.
- ❑ Doğrulama etkili olabilir, ama, örneklenebilecek bilinen bir polinom zamanlı algoritma yoktur.
- ❑ Bu nedenle bir çözüm bulunması hesaplama açısından zorlayıcı olabilir
- ❑ Örnek olarak gezgin satıcı problemi (traveling salesman problem), boole memnuniyeti (boolean satisfiability) olarak verilebilir.

P Problemi ve NP Problemi

Polinom zamanına göre karşılaştırma:

- ❑ P Problemi, etkin bir şekilde çözülebilir
- ❑ NP problemi verimli doğrulanmadığında, etkili bir çözüm olmayabilir.

Zaman karmaşıklıklarına göre karşılaştırma:

- ❑ P problemi polinom zamanlı algoritmalarıdır
- ❑ NP probleminin verimli doğrulama algoritmaları vardır, ancak verimli çözüm algoritmaları olacağı garanti edilemez.

Sonuç olarak

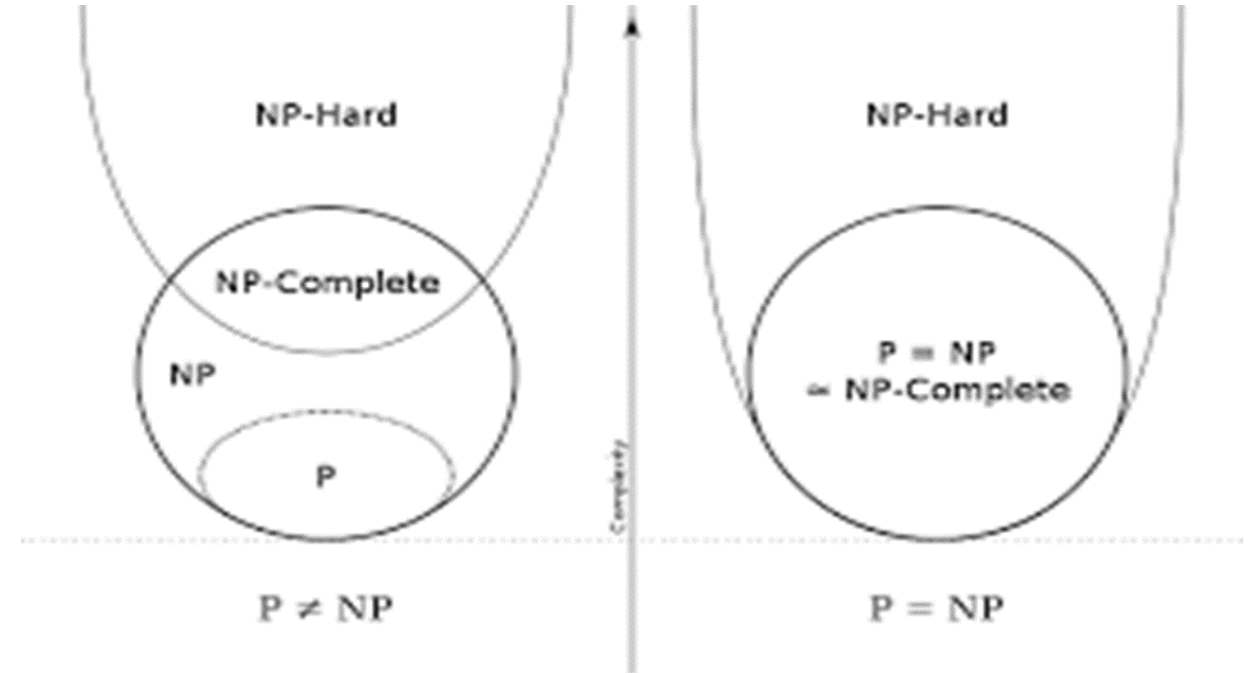
- ❑ Polinom probleminin etkin çözümleri bulunabilir.
- ❑ NP problemi için bir çözüm önerilmişse, bu çözüm etkili olarak şekilde doğrulanabilir.

P ile NP Problemi İlişkili midir?

- ❑ P problemi genellikle karar problemleridir (evet/hayır cevapları)
- ❑ NP problemi ise, karar veya optimizasyon problemleri olabilir.
- ❑ P, NP'nin bir alt kümesi olarak yorumlanır.
- ❑ NP probleminde, NP'nin P'nin öz altkümesi olup olmadığı, bir başka ifade ile eşit olup olmadıkları bilinmez.

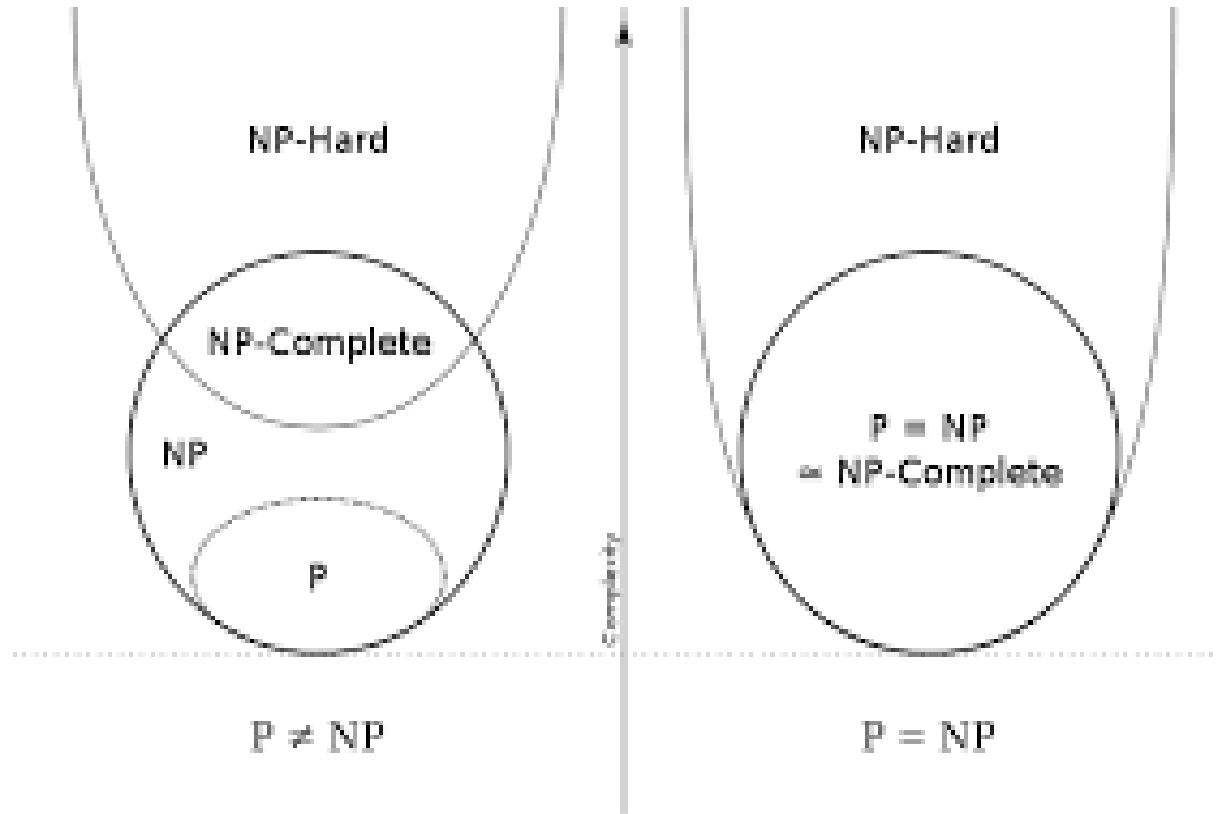
Uygulamada P ve NP Problemi

- ❑ P'deki problemlerin pratikte etkin bir şekilde çözülebileceği düşünülmektedir.
- ❑ NP problemleri hesaplama açısından zordur.
- ❑ NP probleminin çözümü için bilinen etkili bir algoritma yoktur
- ❑ NP probleminin kesin bir çözümü olmadığı için, büyük örneklerde *uygulanamaz* sonucu verir.



Çözülememiş NP Sorusu

- ❑ Bilgisayar biliminin en önemli problemlerden biri P'nin NP'ye eşit olup olmadığıdır.
- ❑ Eğer P, NP'ye eşitse, NP'deki tüm problemler P'de de olacaktır



Turing Complete

Sonsuz belleğe sahip fiziksel bir aygıt yoktur.

FAKAT

Sınırlı belleğin sınırlaması ihmal edilirse, pek çok programlama dili «Turing-complete» olarak değerlendirilir.

İlk Örnekleri:

i) Babbage (1830) Analitik Makine

« conditional loop » ilk defa kullanılmış olduğundan ilk Turing-complete makinedir.

ii) Konrad Zuse (1941) Z3 ve Z4 bilgisayarlar

iii) 1951 Eniac (1945)

HT uygulamalarındaki çeşitli kuramlar

☐ Automata Theory -

- ❖ Game of Life

- ❖ Cellular Automaton (İngiliz matematikçi John Horton Conway 1970)

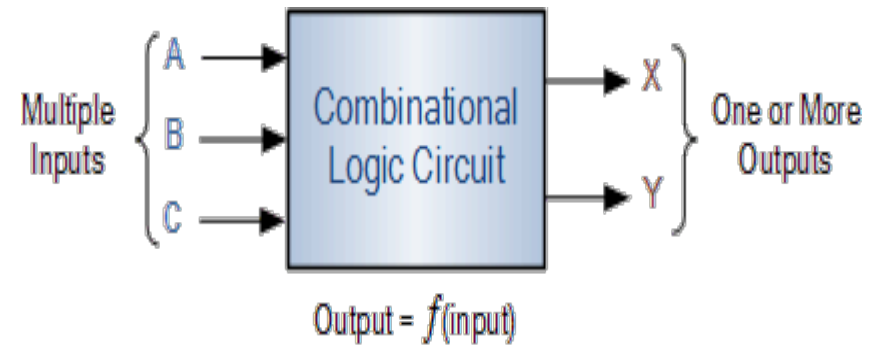
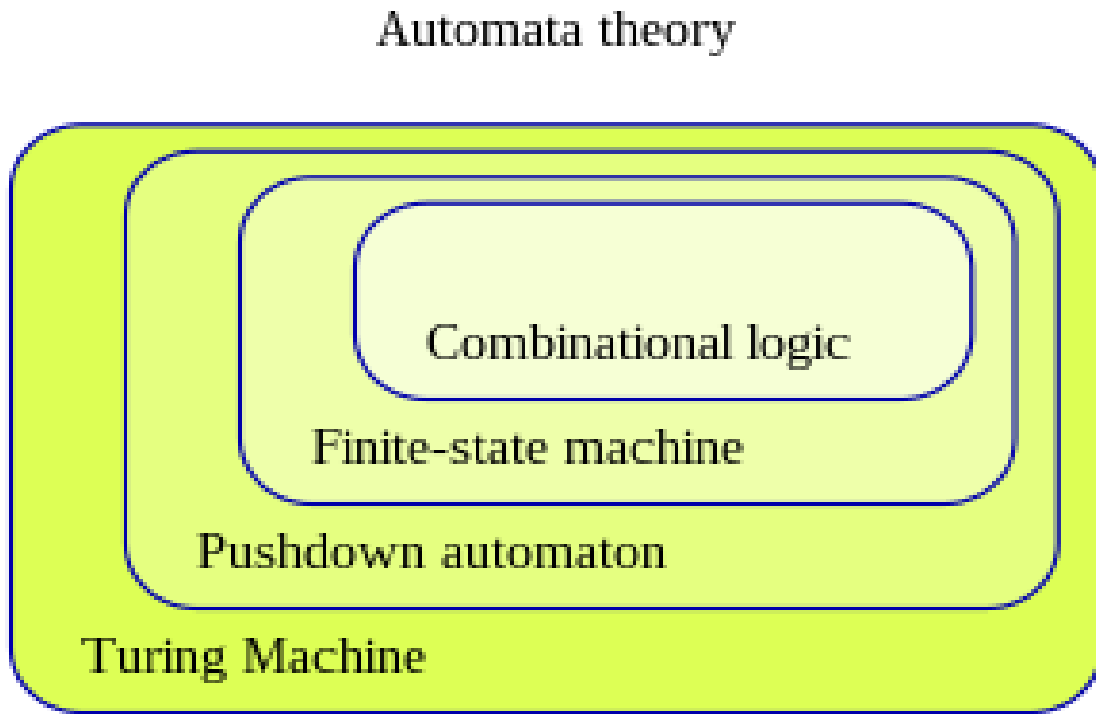
☐ Formal Language

☐ Computability theory

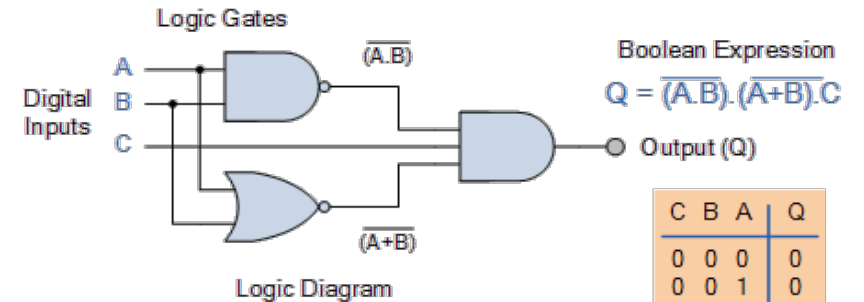
☐ Complexity theory

Otomat Teorisi

Sonlu sayıda durumla Sonlu Durum Otomatları tasarlanır



Boolean Algebra
Truth Table
Logic Diagram

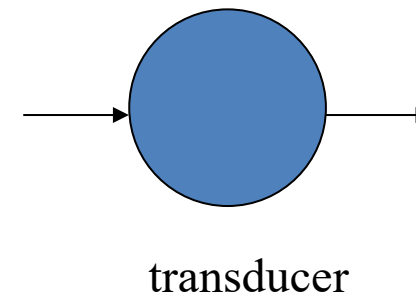
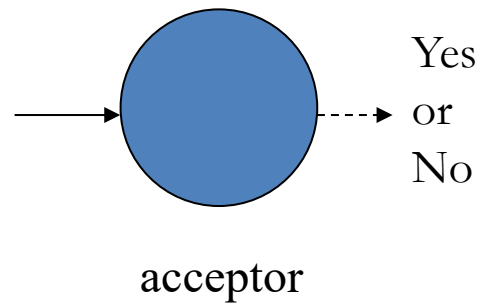
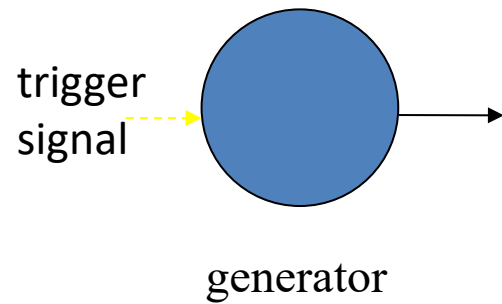
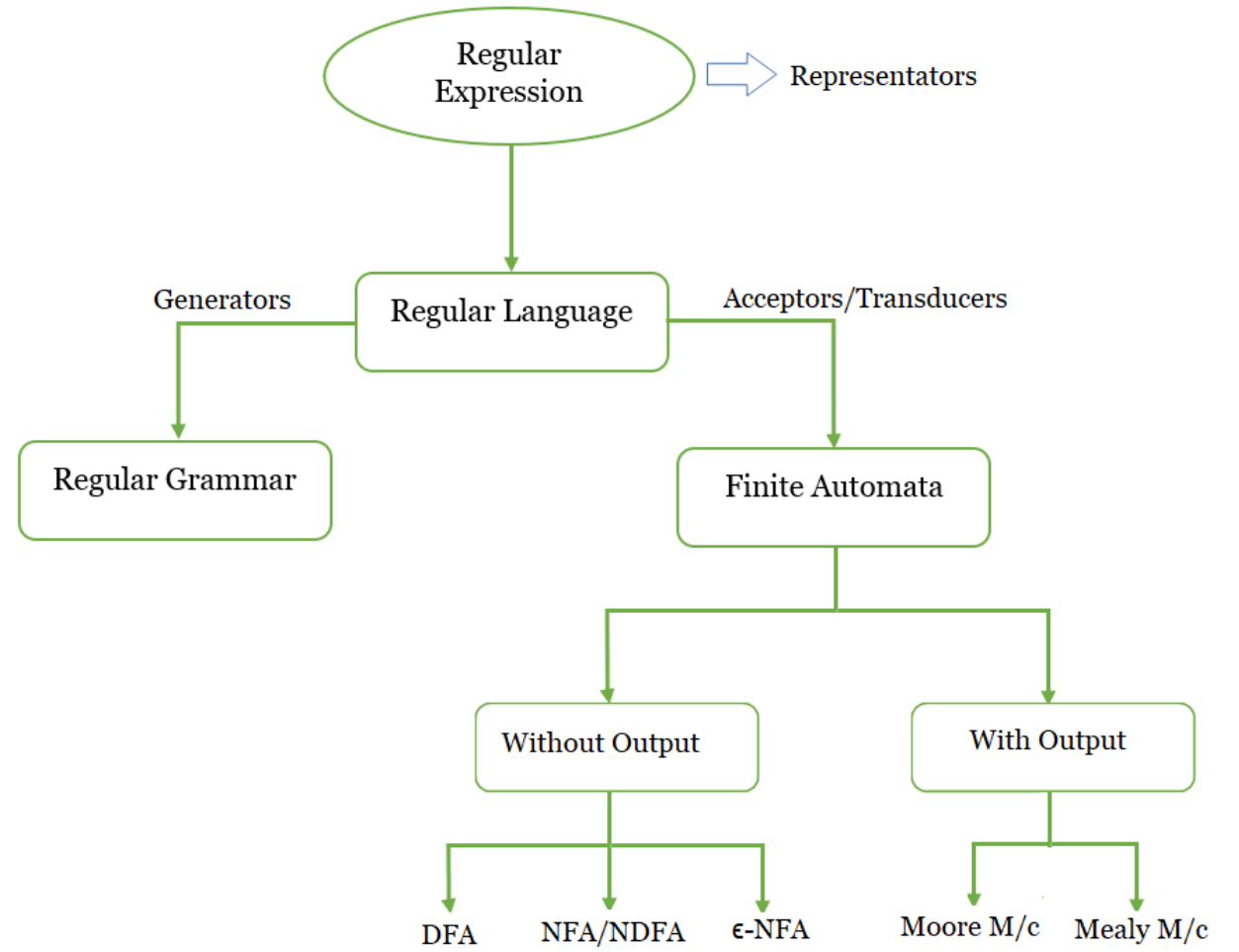


Boolean Expression
 $Q = \overline{(A.B)} \cdot \overline{(A+B)} \cdot C$

Typical Truth Table

C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0

Sonlu Durum Otomatlarının Çeşitleri



Otomata Teorisi Modeli: Üretici / Generator

□ “natural language” grammar

(generating “sentences” spoken by people)

□ reception robot

(speaking organized words and sentences)

□ context-free grammar

(generating strings of symbols)



Reception robot
--- Expo 2005

Otomata Teorisi Modeli: Alıcı / Acceptor

❑ digital lock

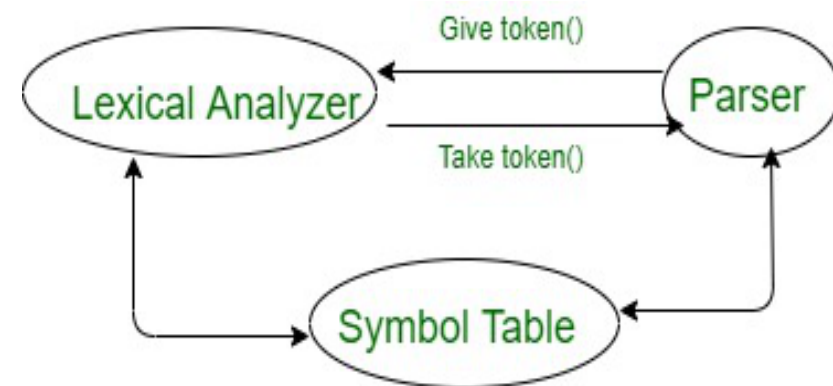
(accepting digits)

❑ lexical analyzer

(recognizing computer language keywords)

❑ finite automaton

(accepting valid strings of symbols)



Otomata Teorisi Modeli: Dönüştürücü /Transducer

□ Interpreter

(translating natural languages)

□ Compiler

(translating high-level languages into machine codes)

□ Turing machine

(transforming strings of symbols)

Otomata Teorisinin Temel Bileşenleri

Sembol /Symbol: These are either individual objects or separate entities. These can be any letter, alphabet or any picture.

Dizgi /String: These are a finite collection of symbols from the alphabet, and are denoted by w .

Dil / Language: A collection of appropriate strings is called a language. A language can be Finite or Infinite.

Otomata Teorisi

□ Bu otomatlar nasıl çalışır?



Vending machine odası
Hokkaido, Japan 2004

Deterministik Sonlu Durum Otomatı (DFA) ile tasarlanır

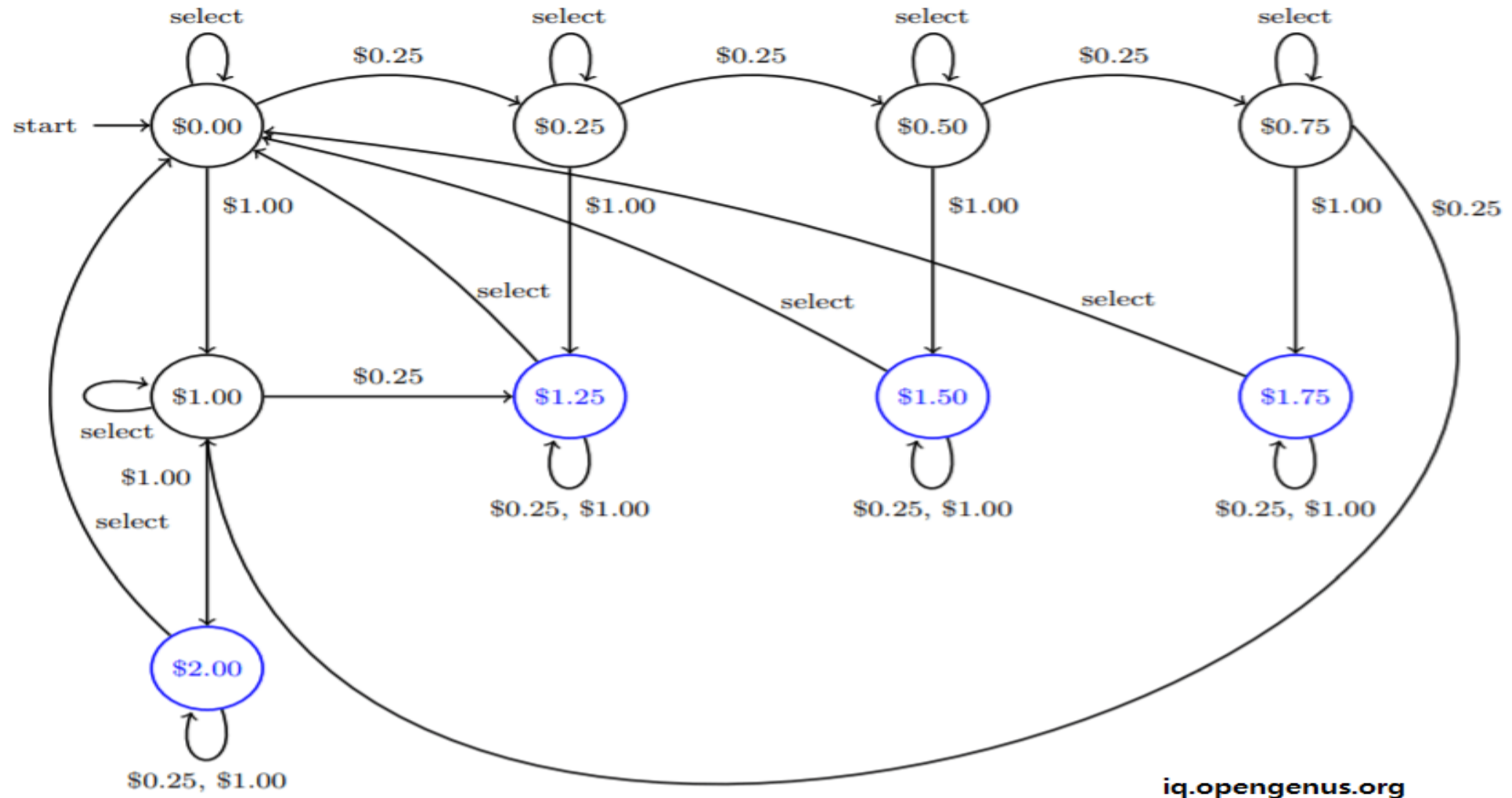
$Q = \{\$0.00, \$0.25, \$0.50, \$0.75, \$1.00, \$1.25, \$1.50, \$1.75, \$2.00\}$ (durumlar)

$\Sigma = \{\$0.25, \$1.00, \text{select}\}$ alfabe

$q_0 = \$0.00$ başlangıç durumu

$A = \emptyset$ kabul durumlarının sonlu kümesi

Vending Machine



Finite Automata (FA): Bazı Uygulama Alanları

Traffic Lights

Video Games

CPU Controllers

Protocol Analysis /Design

Regular Expression Matching

Vending Machines

Speech Recognition

Natural Language Processing

Asynchronous circuits /Digital
Circuit Design

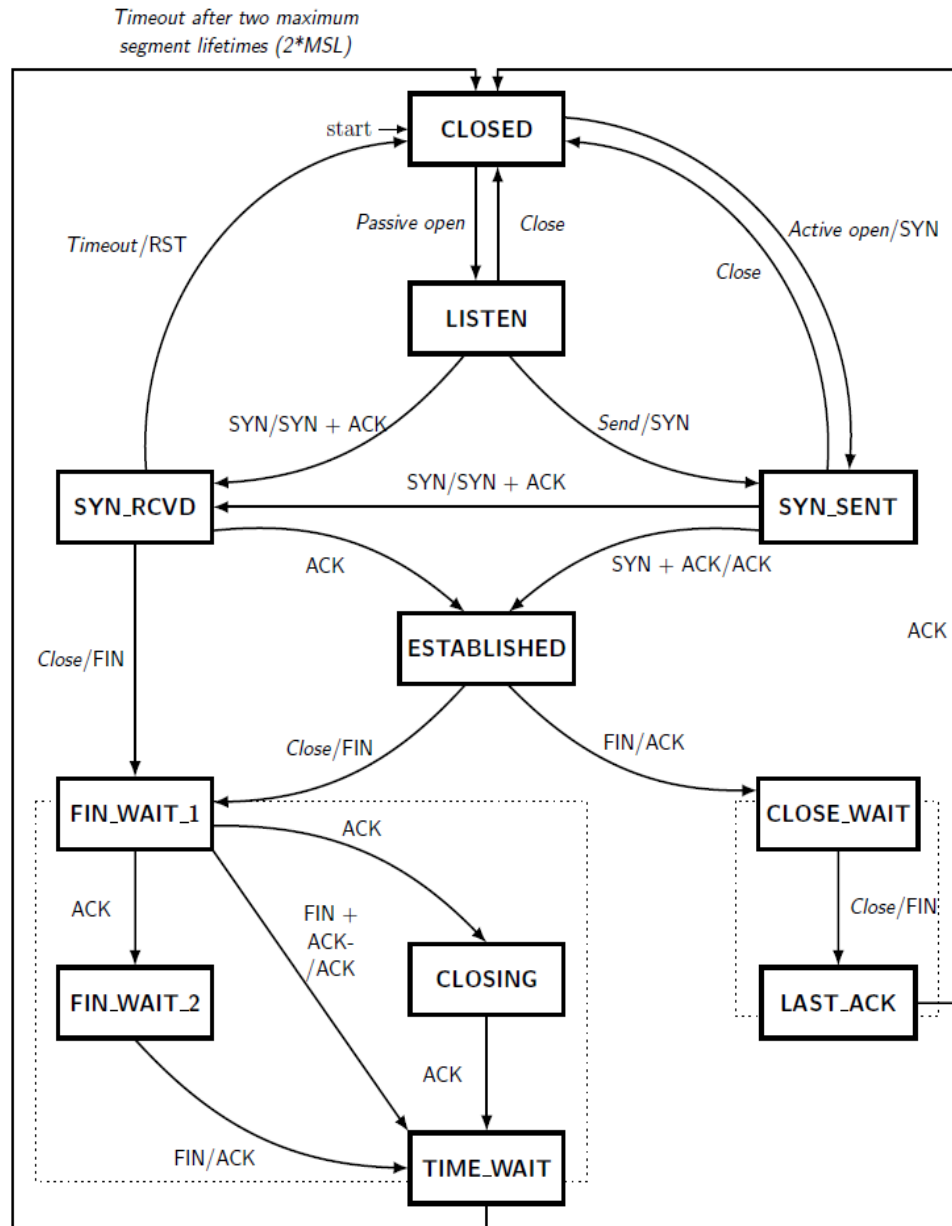
Coding theory,

Concurrent Systems

Software and Hardware

verification

Hardware Testing



Internet Protokolü : TCP 'nin FA ile modellenmesi

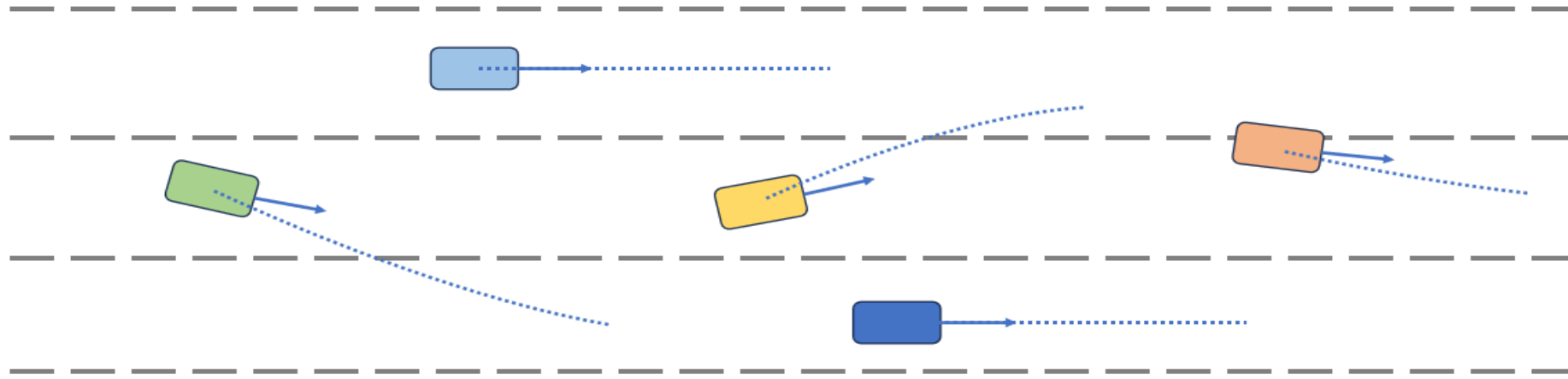
Sıralı Makinelerde FA ile Modellenebilir Problemler

- Asynchronous circuits /digital circuit design
- Coding theory,
- Concurrent systems
- Software and hardware verification
- Hardware testing
- Protocol design

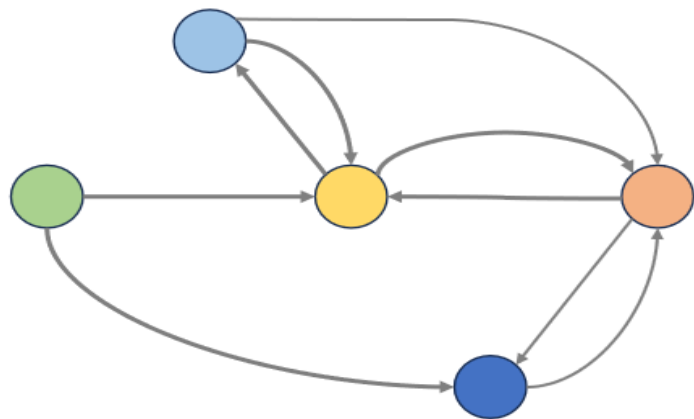
En basit Otomat Örneği: Vending Machines

İçinde çeşitli yiyecek/içecek içeren makineler sonlu durum otomatları ile tasarlanarak çalışma süreçleri kontrol edilir.

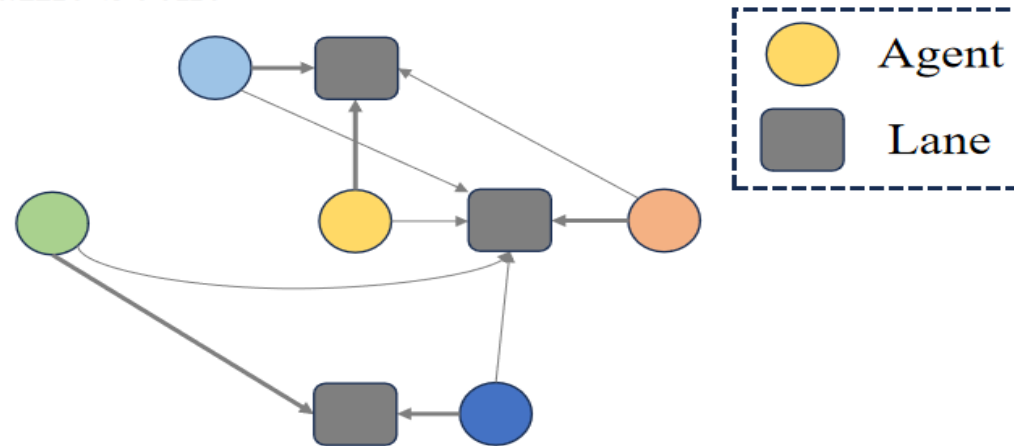
Otomata Örneği: Autonomous Driving (AD)



Traffic Scene



Dynamic Scene Graph

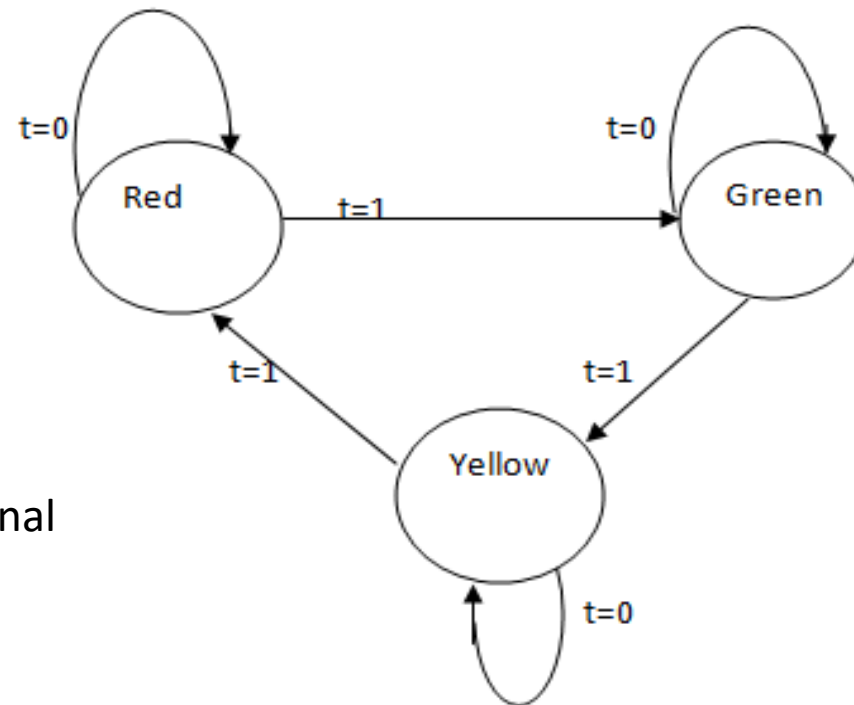


Static Scene Graph

Otomata Örneği: Trafik Işıklarının Otomasyonu

The optimization of traffic light controllers in a city is a systematic representation of handling the instructions of traffic rules.

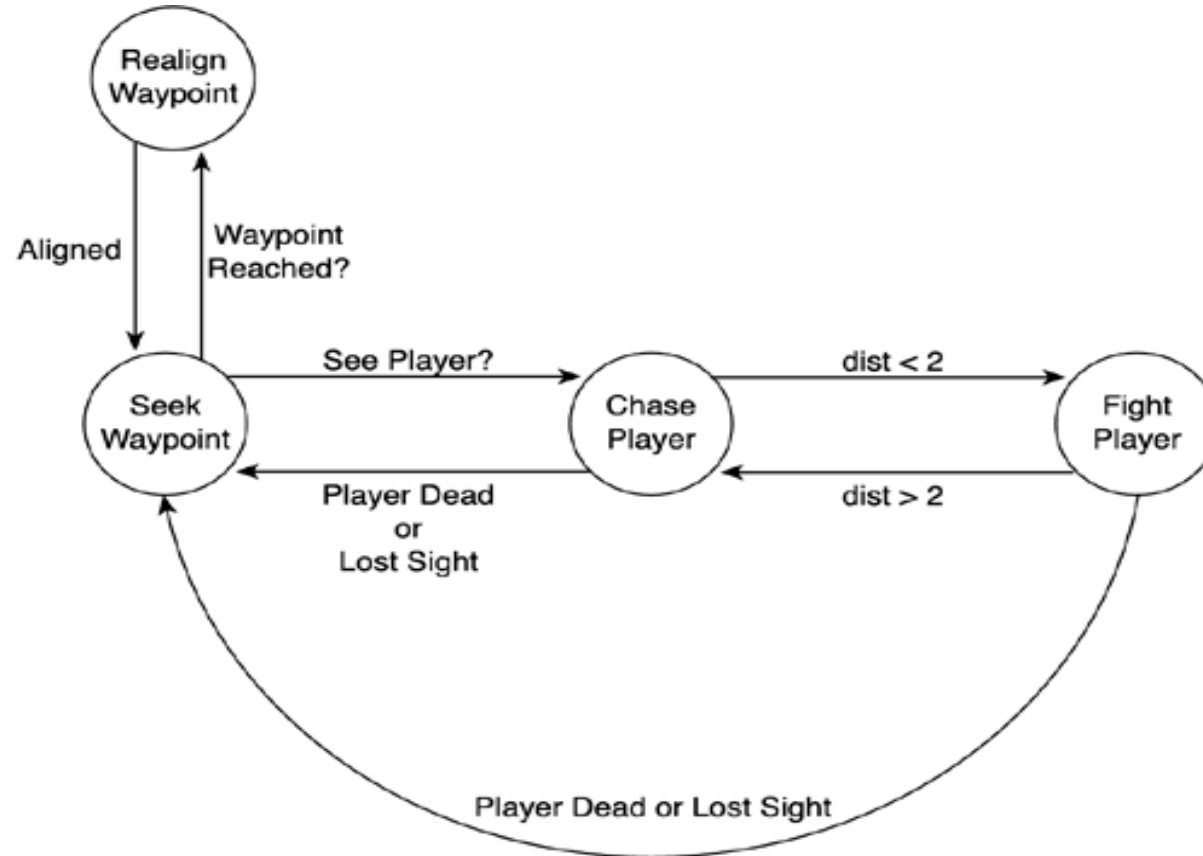
Its process depends on a set of instruction works in a loop with switching among instruction to control traffic



inite state machine for traffic signal

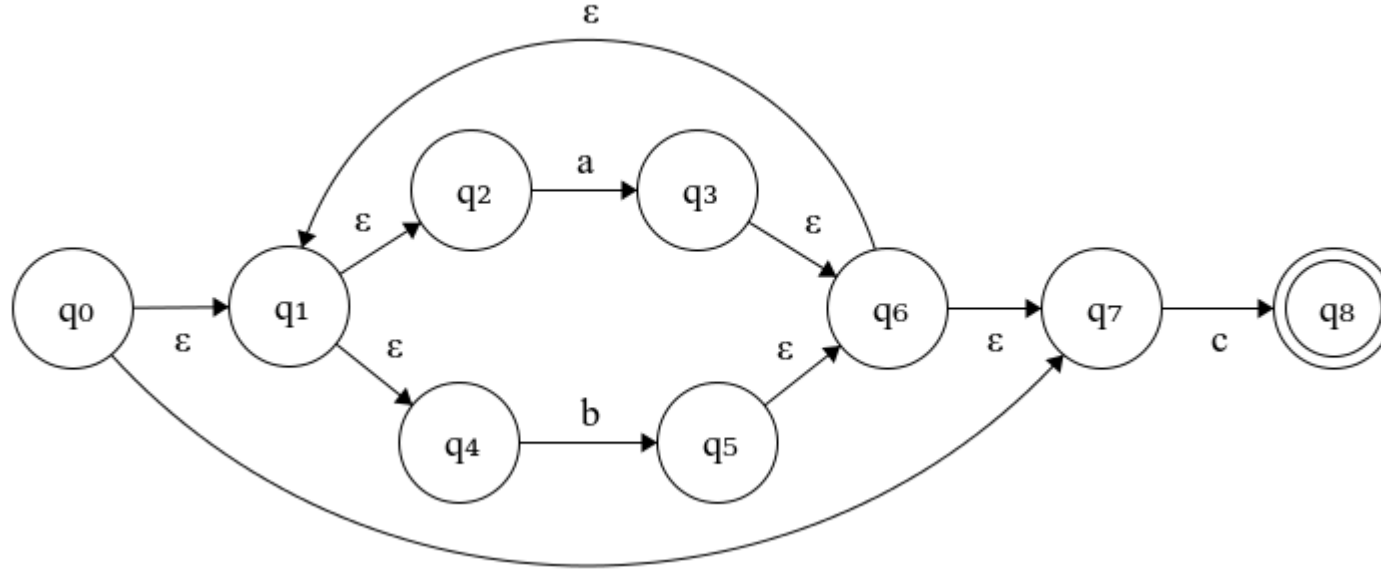
Otomata Örneği

Video Oyunları : Video / bilgisayar oyunlarının farklı düzeyleri otomatanın durumları olarak modellenir.



Otomata Örneđi

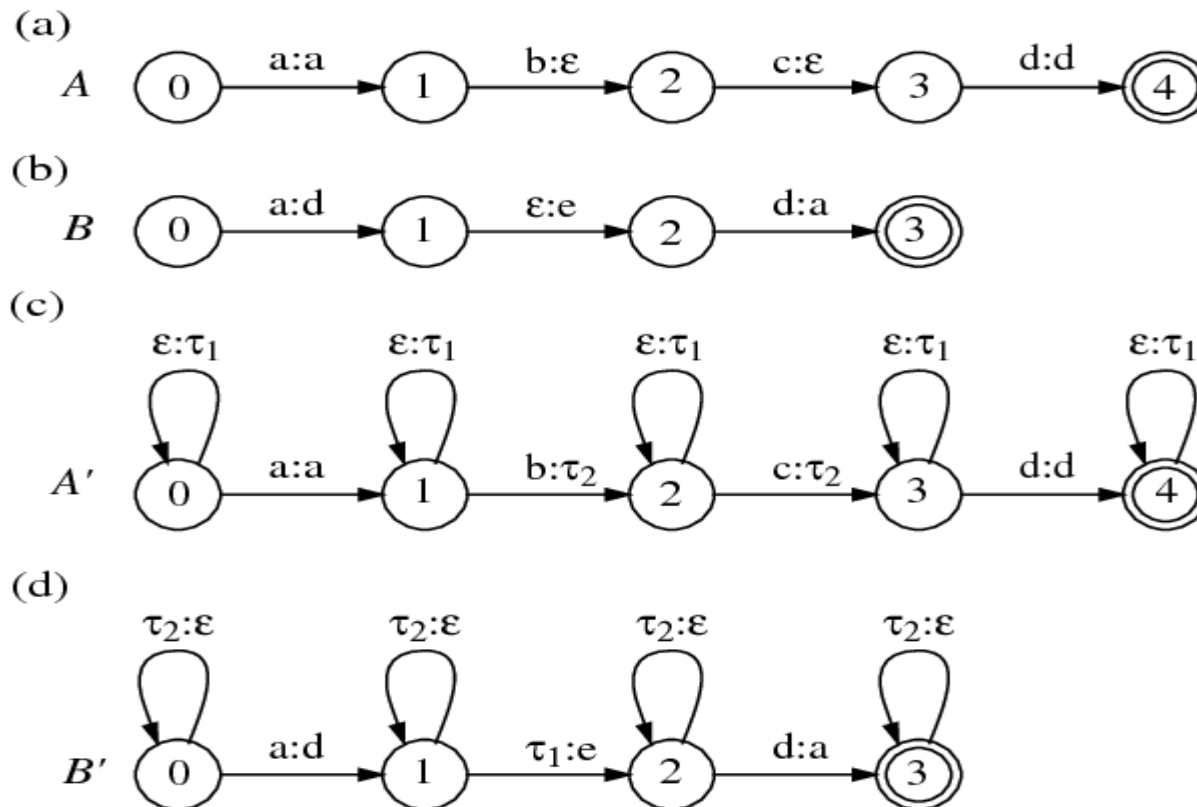
Metin Çözümleme / Text Parsing: Metin içerisinde alınan bir dizginin (string) tanımlanmış dilbilgisi kurallarına uygunluđunun kontrolü



Düzgün ifadelerin eşleşmesi: İki ya da daha fazla düzgün ifadenin birbirine denk olup olmadığını kontrol eden bir tekniktir.

Otomata Örneği: Konuşma Tanıma

□ Sözcüklerin ve deyimlerin makine-tarafından okunabilir forma dönüştürmede kullanılan bir teknolojidir.



Özet: HT ile İlişkili Bazı Uygulama Alanları

Fields	Related theory
Compiling theory	Formal languages
Switching circuit theory	Automata theory
Algorithm analysis	Computational complexity
Natural language processing	Formal languages
Syntactic pattern recognition	Formal languages
Programming languages	Formal languages
Artificial intelligence	Formal languages and automata theory
Neural networks	Automata theory