



# **YZM 3217 – YAPAY ZEKA**

## **DERS#2:AJANLAR (AGENTS)**

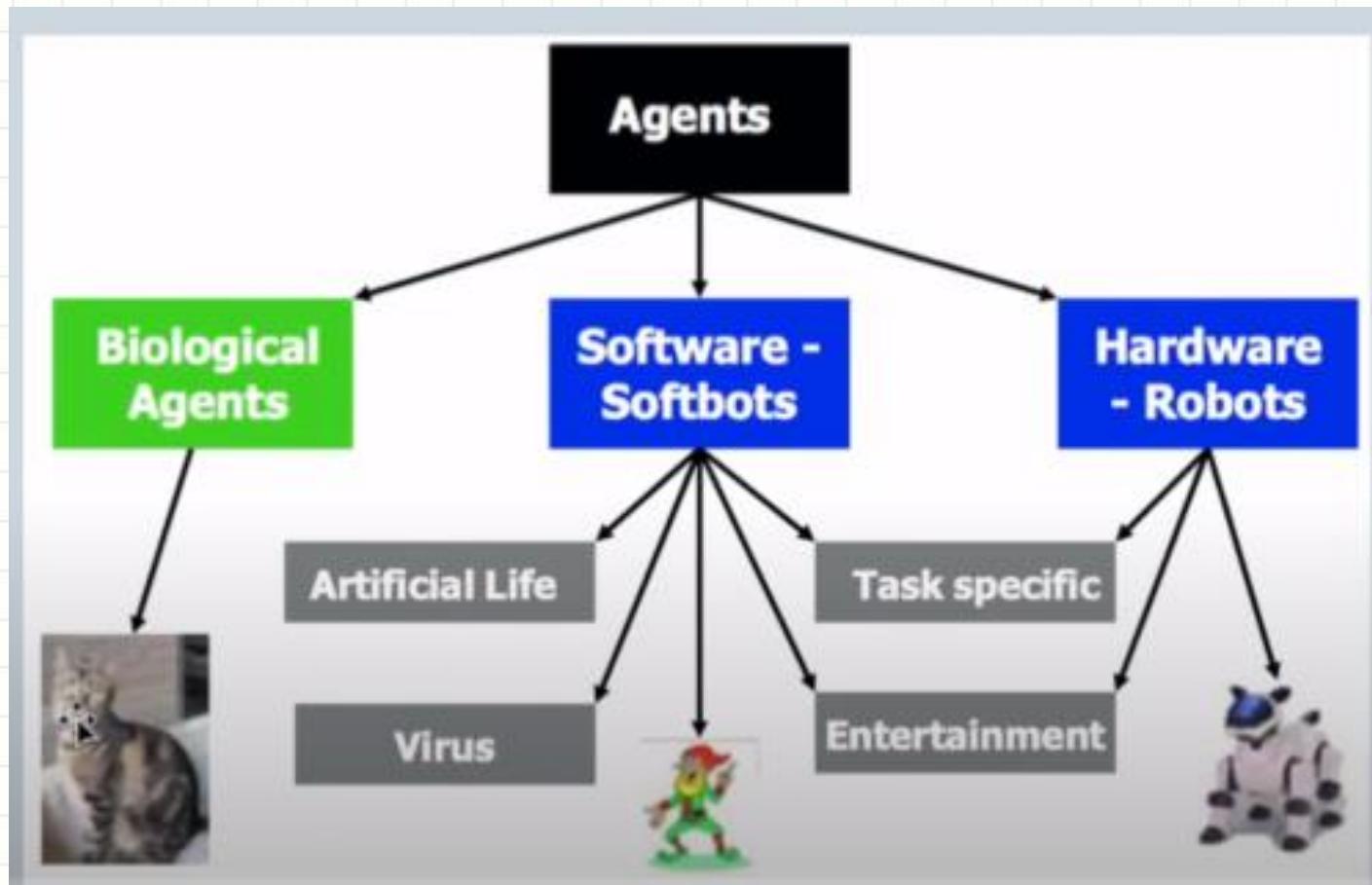
# Agents

- **Agent:** An object that receives environmental information with its **sensors** and **reacts** to this environment with its reactants.
- The agent's perceptual inputs at any given time are called percepts.
- **Human agent:**
  - Sensors: eyes, ears,...
  - Reacts: feet, hands, mouth...
- **Robot agent:**
  - Sensors: camera, infrared finders,...
  - Reacts: various motors, mechanical arms,...

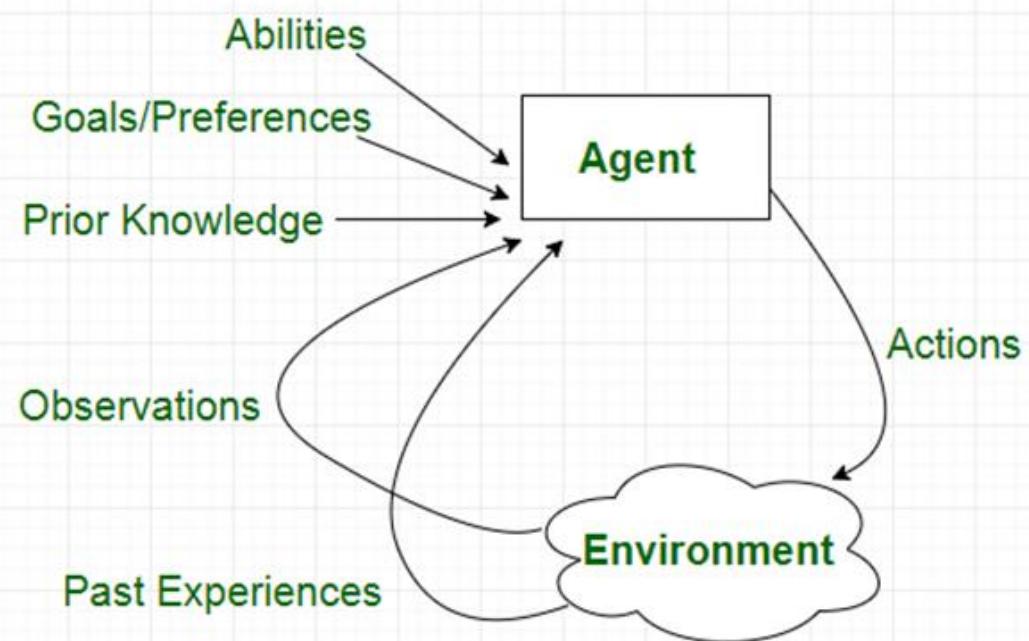
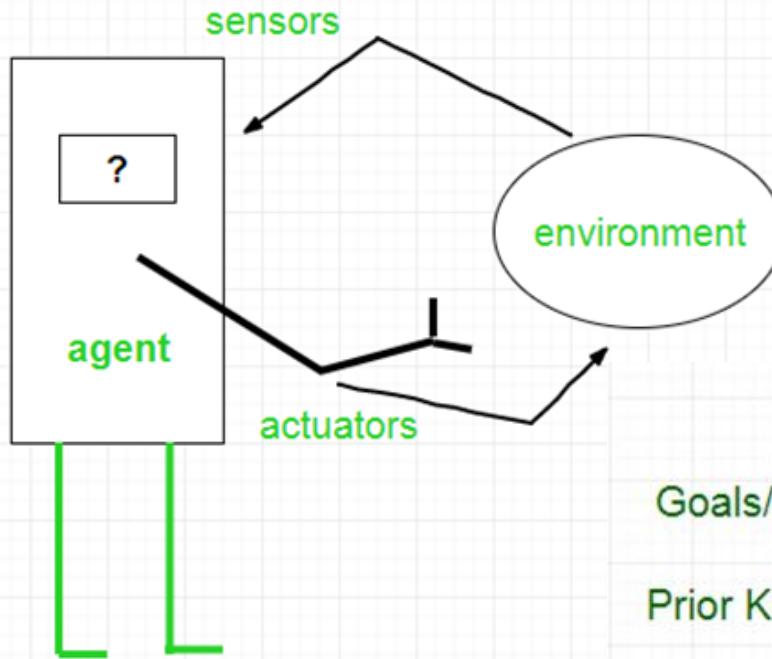
# Ajanlar

- **Ajan:** **Algılayıcıları** ile **ortam** bilgilerini alan ve **tepki vericileri** ile bu ortama tepki veren nesne
- Ajanın herhangi bir zamandaki algısal girdilerine **duyum (percept)** adını verilir
- **İnsan ajan:**
  - **Algılayıcılar:** gözler, kulaklar,...
  - **Tepki vericiler:** ayaklar, eller, ağız...
- **Robot ajan:**
  - **Algılayıcılar:** kamera, kızılıötesi bulucular,...
  - **Tepki vericiler:** çeşitli motorlar, mekanik kollar,...

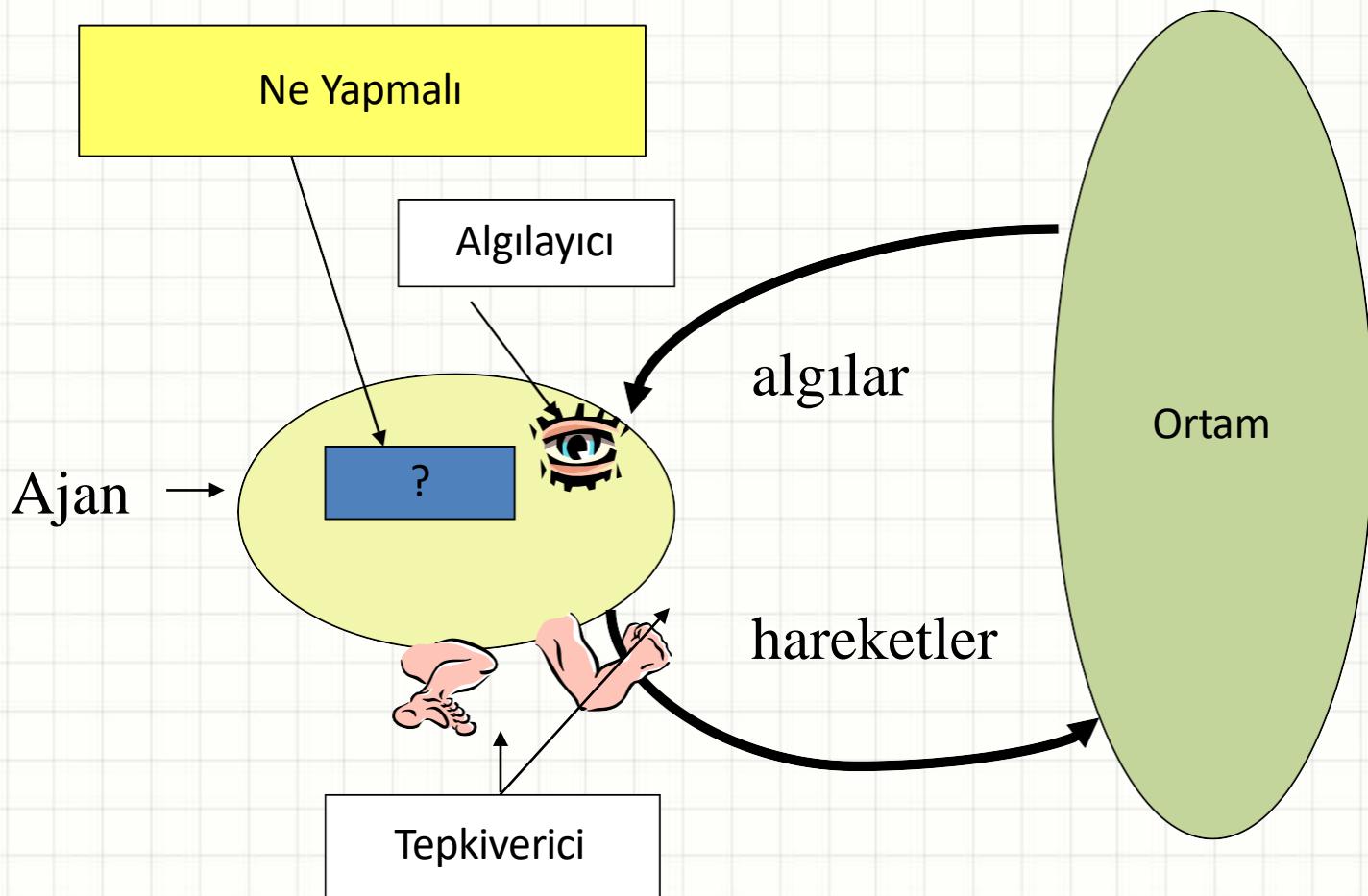
# Ajanlar-Agents



# Ajanın yapısı-Agent structure



# Ajanın yapısı



# Ajanların Tasarımı-Design of Agents

- **Agent program**
  - Function that transforms perceptions into movements
- **Architectural**
  - Computerized system in which programs are executed
- **Agent = architecture + program**
- To design the agent program, we must have knowledge of the following:
  - Possible detections and movements
  - Goals and measure of success
  - The type of environment in which the agent will operate

# Ajanların Tasarımı

- Ajan programı
  - Algıları hareketlere dönüştüren işlev
- Mimari
  - Programların yürütüldüğü bilgisayarlı sistem
- Ajan = mimari + program
- Ajan programını tasarlamak için aşağıdakiler hakkında bilgimiz olmalıdır:
  - *Olası algılamalar ve hareketler*
  - *Amaçlar ve başarı ölçütı*
  - *Ajanın işlem yapacağı ortamın türü*

# Ajanlar ve Ortamları

Ajan fonksiyonu, ajanın algıladığı bilgileri ( $P$ ) hareketlere ( $A$ ) dönüştürür:

$$F(P) \rightarrow A$$

- Ajan programı, ajanın fiziki mimarisi üzerinde  $F(P)$ 'yi üretmek içindir
- Bir ajan, alıcılarından gelen bilgilere ve tepki vericiler ile yapabileceklerine göre doğru hareketi yapmaya yönelmelidir.

# Ortam Türleri

- Ajanın tasarımı için önemlidir
- Tam/kısmi gözlemlenebilir (Fully/Partially observable)
  - Zamanın verilmiş diliminde ajanın algılayıcıları tarafından ortamın tüm durumunun gözlemlenmesi mümkün mü?
- Belirgin (deterministik)/Tahmini (stokastik)
  - Deterministik ortam: Çevrenin bir sonraki durumu, mevcut durum ve ajanın hareketleri tarafından belirlenebilir
- Episodic (aralıklı, bağlantısız, bölümlü)/Ardışık (sıralı)
  - Ajanın yaşamı küçük parçalara (epizotlara) bölünür. Epizottaki hareketin seçimi, yalnız epizodun kendisine bağlıdır.
  - Epizot önceki epizotlara bağlı değil

# Ortam Türleri (*devamı*)

- Statik (durağan)/Dinamik
  - Ajanın karar oluşturduğu (düşündüğü) süreç içinde ortam değişirse bu dinamik ortamdır, değişmezse statik ortamdır.
  - *Yarı-dinamik*: ortam değişmez, ama ajanın başarı ölçüsünün değiştiği durum
- Ayrık/Sürekli
  - Algılamalar veya hareketler sayılabilirse (ayırık ise) ortam ayrık, sayılamıyorsa süreklidir
  - Satranç ortamı ayrıktır, çünkü her hamlede sınırlı sayıda hareketler var.
- Tek/Çoklu ajan
  - Tek ajan ortamda kendi başına hareket eder. Çoklu ajanlar işbirliği yaparlar
  - Günlük hayatın problemlerinin bir çoğunda kısmı gözlemlenebilir, deterministik, ardışık, dinamik, sürekli ve çoklu ajanlı bir ortam yapısı görülür.

# TAMAMEN GÖZLEMLENEBİLİR ORTAM

- Ajan sensörler ile etrafını çevreleyen tüm alanı birim zaman içinde tamamen gözlemlayebilmesidir.
- **Örneğin;**
- SATRANÇ: Satranç tahtası tamamen gözlemlenebilir ve böylece rakibin hamleleri de tamamen gözlemlenebilir.
- SÜRÜŞ: Çevre kısmen gözlemlenebilir çünkü köşeyi dönünce ne olduğu bilinmiyor. Bundan dolayı bu ortam tamamen gözlemlenemez.



# STATİK, YARI DİNAMİK, DİNAMİK

- **Statik:** Ortam değişmez
- **Yarı Dinamik:** Ortam değişmez ama zamanla ajanın performansı değişir. Örneğin Satranç oynarken zamanla ortam değişmez, ortam sabittir fakat zamanla oyuncuların performansı değişir.
- **Dinamik:** Ajan çalışırken ortamın değişir



# DETERMINİSTİK ve STOKASTİK

- **Deterministik:** Ortamın bir sonraki durumunu ajan tarafından yürütülen faaliyet belirler. Yani bir sonraki durum şimdi yapılan işten etkilenederek gelişir.
- **Stokastik:** Stokastik ortam rastgeledir ve ajan tarafından tamamen belirlenemez.
- **Örneğin;**
- Deterministik ortam için mevcut durumda kapı açıkken eğer kapıyı itersen kapı kapanır. Yani şuan yapılan itme faaliyeti bir sonraki durumu etkiler ve kapı kapanır.
- Stokastik ortam için otonom giden arabaların hareketleri zamanla ortama bağlı olarak değişir.

# TEK AJAN ve ÇOKLU AJAN

- **Tek Ajan:** Ortamda yalnızca bir ajanın olmasıdır.
- **Örneğin:** Labirentte yalnız bırakılan bir kişi, tek ajana örnektir.
- **Çoklu Ajan:** Ortamda birden fazla ajanın olmasıdır.
- **Örneğin:** Futbol maçında her takımda 11 oyuncu olması.
  
- Çoklu Ajanlar **Rekabetçi** ve **İşbirlikçi** olarak faaliyet gösterebilir.
- **Örneğin;** futbol maçında her iki takımın birbirine üstünlük kurmaya çalışması ‘Rekabetçi’, takımın kendi içerisinde başarıyı sağlama<sup>6</sup>na yönelik çalışma ise ‘İşbirlikçi’ faaliyetlere örnektir.

# AYRIK ve SÜREKLİ

- **Ayrık:** Bir ortam, çıktıyi elde etmek için çevrede düşünülecek sınırlı sayıda eylemden oluşuyorsa, buna ayrık bir ortam denir.
- **Örneğin;** Satranç oyunu, yalnızca sınırlı sayıda hamleye sahip olduğu için ayrıktır. Her oyunda hamle sayısı değişebilir, ancak yine de sonludur.
- **Sürekli:** Yapılan işlemlerin yapıldığı ortam numaralandırılamaz, yani kesikli değilse sürekli olduğu söylenir.
- **Örneğin;** Kendi kendine giden arabalar, eylemleri numaralandırılamayan sürüş, park etme vb. olduğundan sürekli ortamlara bir örnektir.

# BÖLMELİ ve ARDIŞIK

- **Bölmeli:** Ajan faaliyetlerine bölmelere ayırarak gerçekleştirir. Ajan her bölümü algılayıp ardından tek işlem gerçekleştirmesidir.
- **Örneğin;** Fabrikalarda ki bantlarda her işlem böülümlere ayrıılır ve herkes bir iş yapar. Ayrıca banttaki parçaların hatalı olup olmasına karar vermek için geçmiş kararları incelemeye gerek yoktur.
- **Sıralı:** Ajan faaliyetlerini sıralı olarak sürdürür.
- **Örneğin;** Hastaneye başvuran hastanın hastalığının geçmişten günümüze sıralı olarak incelenmesiyle bulunur.



# Ajanın Başarı Kriteri

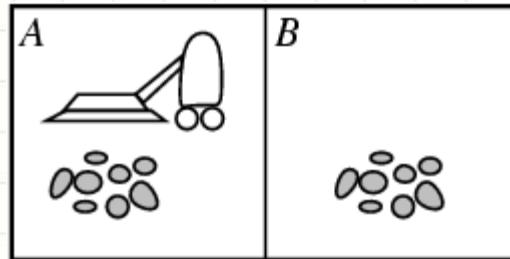
1. Başarı kriterini tanımlayan performans ölçüsü
2. Ajanın çevreyle ilgili ön bilgisi
3. Ajanın gerçekleştirebileceği eylemler
4. Ajanın bugüne kadarki algılama dizisi



# Agent success criteria

- Performance measure that defines success criteria
- Agent's prior knowledge of the environment
- Actions that the agent can take
- Agent's detection sequence to date

# Örnek: Süpürge ajanının dünyası



- **Algılamalar:** mekan ve içeriği, örn: [A,Kirli]
- **Hareketler:** Sol, Sağ, Süpür, İşlem Yok

# Bir Ajanın Dış Nitelikleri

- Bir ajanın belirli bir zamana kadar algıladığı her şey, ajanın **algı serisini (percept sequence)** oluşturur.
- Matematik bakış açısıyla bir ajanın davranışı, herhangi bir algı serisini bir eyleme eşleyen **ajan fonksiyonu** (agent function) ile tanımlanır.
- Ajanı betimleyen ajan fonksiyonunu bir tablo ile gösterebilir. Çoğu ajan için, eğer algı serisinin uzunluğuna bir **sınır** getirmezsek, bu tablo çok büyük, hatta sonsuz büyüklükte, olacaktır.

# Elektrik süpürgesi dünyası için basit bir ajan fonksiyonunun kısmi tablolAŞtırılması

## ALGI SERİSİ (percept sequence)

[A, Temiz]

[A, Kirli]

[B, Temiz]

[B, Kirli]

[A, Temiz], [A, Temiz]

[A, Temiz], [A, Kirli]

...

[A, Temiz], [A, Temiz], [A, Temiz]

[A, Temiz], [A, Temiz], [A, Kirli]

## EYLEM-Action

Sağ

Temizle Sol

Temizle Sağ

Temizle

...

Sağ Temizle

# Algı Serisinden Eyleme İdeal Eşleme

- Ajanın davranışları yalnız algı serisine bağlı ise olası tüm algı serilerine karşı gelen eylemler tablo haline getirilerek bir ajan tanımlanabilir.
  - Çoğu zaman bu tablo **çok uzun** olacaktır.
- Oluşturulan tabloya "**algı serisinden eyleme eşleme**" denir.
- Eğer eşleme ajanı tanımlıyorsa ideal eşleme de ideal ajanı tanımlar.
- Eşleme için tablonun her bir elemanınının ayrı ayrı belirtilmesi gerekmez
  - Yazılacak bir **program** ile de ajan tanımlanabilir

# Örnek Eşleme

- **Basit ajan:** karekök fonksiyonu
- **Algı serisi:** Hesap makinasının tuşları
- **İdeal eşleme:** Girilen pozitif sayı  $x$  ise  $z^2 \approx x$  olacak şekilde 4 basamak doğrulukta  $z$ 'yi göstermek
- Tablo yerine Newton yöntemi kullanılarak yazılan program ile ajan tanımlanabilir.
- Tablo çok uzun olmasına karşın **ajan çok kısa** bir programdır.

Tablo

Algı X	Eylem Z
1.0	1.0000
1.1	1.0488
1.2	1.0954
1.3	1.1401
...	

Program

```
Function SQRT(X)
z←1.0 /* ilk tahmin */
repeat until |z2-x|<10-4
    z←z-(z2-x)/(2z)
end
return z
```

# Neden arama tablosu başarısızdır

- Ajan programı yazmanın en basit yolu tablo kullanmaktır (**look-up table**)
  - Bu durumda olası tüm algı serisinin **bellekte** tutulması ve **indeks** kullanarak erişilmesi gereklidir
- Tablo kullanımında aşağıdaki olumsuzluklar ortaya çıkar:
  - Oldukça fazla kayıt gereklidir
    - Satranç oynayan ajan için  $35^{100}$  kayıt
  - Tabloyu oluşturmak çok zaman alır
  - Özerklik yoktur
    - Ajanın tüm hareketleri önceden belirlenmiştir (**kaydedilmiştir**)
  - Ajana bir derece özerklik tanınarak öğrenme mekanizması oluşturulsa bile tüm girişler için **tablonun doğru değerlerini bulması sonsuza kadar sürer**

# Ajan Programı

```
function Ajan(algı) returns hareket
    bellek ← Bellek_Güncelle(bellek, algı)
    hareket ← En_iyi_Hareket(bellek)
    bellek ← Bellek_Güncelle(bellek, hareket)
return hareket
```

# Ajan Programları - Arama tablosu

```
function Tablo_Tabanlı_Ajan(algı)  returns hareket
    algılar, /*başlangıçta boş */
    tablo /* algılar üzere indekslenmiş tablo, başlangıçta tam
belirlenmiştir*/
    algı'yı, algılar tablosunun sonuna ilave et
    hareket ← Tabloya_Bak(algı, tablo)
return hareket
```

# Mantıklı Ajanlar (Logical Agents)

- Algılayabildiklerine ve yapabildiklerine dayanarak “**doğru şeyler yapmak(do things right)**” için çaba gösteren ajanlar
- **Akıllı (rasyonel) bir ajan** doğru şeyi yapandır: tablosunda her giriş doğru olarak doldurulmuştur.
- Doğru hareket ajanın başarılılık etkenlerindendir
- **Başarı ölçüsü(success criterion):** ajanın davranışının doğruluğunu gösteren kistas
  - *E.süpürgesi ajanın başarısı temizlediği kir miktari, harcanan zaman, tüketilen enerji, çıkardığı gürültü(noise) ile, vs. belirlenebilir*
- Mantıklı ajan, her bir algılama için olası hareketler içinden başarıyı **en fazla yapanı** seçmelidir.

# Mantıklılık ve her şeyi başarmak

- Mantıklı olmak her şeyi başarmak değildir(Being rational is not doing everything.)
  - “Her şeyi başarmak” ajanın, tüm hareketlerinin ne ile sonuçlanacağını bildiği anlamına geliyor
  - **Örnek:** caddeyi geçerken, başına pencerenin düşeceğini önceden bilmek ☺
- Mükemmelliğe gerek yoktur(There is no need for perfection)
  - Her zaman doğru şeyler yapmak mümkün değildir
  - Beklenen başarı, nelerin algılandığına bağlıdır (**Örn,** girdi hatalı ise?)
- Mantıksal davranış aşağıdaki etkenlere bağlıdır(**Rasyonel behavior depends on the following factors**)
  - Başarı ölçüsü
  - Algılama tümcesi: ajanın belirli bir anda algıladığı her şey
  - Ortam hakkında ajanın bildikleri
  - Ajanın yapabileceği olası hareketler(Possible actions the agent can take)
- Ajanın davranışları yalnız onun kendi deneyimi, öğrenme ve uyum sağlayabilme yeteneği ile belirlenirse, ajan **özerktir** (**An agent is autonomous if her behavior is determined only by her own experience, ability to learn and adapt**)

# İdeal Ajanlar

- İdeal ajanı oluşturma
  - Olası tüm algılama tümcelerinin harekete dönüştürülmesi (**haritalanması**)
  - İdeal dönüştürme: Her algılama tümcesi için belirlenmiş en yüksek başarı ölçüsüne uygun cevabin bulunması
- Tabloya her zaman gerek duyulmaz
  - Gereken tek şey dönüştürmenin **tanımlanmasıdır**
  - **Örnek:**  $15'$ in karekökünün hesaplanması için büyük tablo yerine küçük bir program yeterlidir

# Ideal Agents

- Creating the ideal agent
  - Converting all possible detection sentences into action (mapping)
  - Ideal conversion: Finding the appropriate answer for each detection sentence with the highest success criterion determined
- Table not always needed
  - All that is needed is to define the conversion
  - Example: Big table for calculating the square root of 15, A small program is sufficient instead.

# İdeal Ajanlar - Özerklik

- Eğer ajanın tüm hareketleri yalnız önceden kaydedilmiş bilgilere dayanırsa, bu ajan **özerk degildir**
- Ajanın özerklik davranışları onun deneyimi ile belirlenmelidir
  - Bazı başlangıç bilgileri ve öğrenme yeteneği gerekmektedir
- Özerk olmayan ajanlar esneklikten yoksundur

# İdeal Ajanlar - Özerklik

- If all actions of the agent are based on pre-recorded information alone, this agent is not autonomous.
- The agent's autonomy behavior should be determined by his experience.
  - Some initial knowledge and learning ability required
- Non-autonomous agents lack flexibility

# Özerklik

- Özerk kelimesi, burada insanın doğrudan kontrolü altında olmayan anlamındadır (here means not under the direct control of man)
  - Örnek: özerk kara aracı (insansız-unmanned)
- Özerk olan ajanlar çevre koşulları değiştiğinde yeni koşullara adapte olarak görevini başarı ile sürdürbilir(Autonomous agents can successfully continue their task by adapting to new conditions when environmental conditions change.)
- Eğer sadece önceden verilen bilgileri kullanırsa **başarısız** olma olasılığı yüksektir.(If it only uses the previously given information, It is likely to fail.)

# Ön Bilgiye Karşı Öğrenme(Learning Prior Knowledge)- I

- Başarılı ajanlar, ajan fonksiyonunun hesaplanması işini üç farklı aşamaya ayırır:(Successful agents divide the work of calculating the agent function into three distinct phases)
  1. ajan tasarılanırken, hesaplamanın bir kısmı tasarımcıları tarafından yapılır(while designing the agent, some of the computation is done by its designers)
  2. ajan bir sonraki eylemini düşünürken daha fazla hesaplama yapar(agent does more calculations as it considers its next action)
  3. ve tecrübeden öğrenmesi sebebiyle davranışını değiştirmek için daha da fazla hesaplama yapar(and because it learns from experience, it makes even more calculations to change its behavior.)
- Akıllı bir ajan, kısmi veya yanlış ön bilgiyi telafi edecek şekilde öğrenebilmelidir:(An intelligent agent should be able to learn in a way that compensates for partial or incorrect prior knowledge)

=> **Özerklik- autonomy**

# Ön Bilgiye Karşı Öğrenme(Learning Prior Knowledge)- II

- Yapay zekalı bir ajana, bir miktar ön bilgiyle birlikte öğrenme yeteneği verilmesi mantıklı olacaktır.(It would make sense to give an AI agent the ability to learn along with some prior knowledge.)
- Çevresi hakkında yeterli tecrübe edindikten sonra, mantıklı bir ajanın davranışı ön bilgisinden **bağımsız** hale gelebilir(After gaining sufficient experience of her environment, the behavior of a rational agent can become independent of her prior knowledge.)

**=> Tam özerklik (full autonomy)**

- Öğrenmenin işin içine katılması, farklı ortamlarda başarılı olacak bir mantıklı ajanın tasarılanmasına olanak veririr(The involvement of learning allows designing a rational agent that will be successful in different environments.)

# Ajanın Nitelendirilmesi-Agent Qualification

- Mantıklı ajanların tasarıımı için PEAS bileşenleri belirlenmelidir(PEAS components must be determined for the design of logical agents)
- PEAS bileşenleri(PEAS components): Bir ajanın hareketi sonucu oluşan sonucun objektif değerlendirmesi için gerekli bileşenler(Components required for an objective assessment of the outcome as a result of the action of an agent)
  - Başarım ölçütü - Performance measure
  - Ortam - Environment
  - Tepkivericiler - Actuators
  - Algılayıcılar - Sensors

# Ajan Programları - Örnek

- Performance measure: Safe, fast, legal, comfortable trip, maximize profits
- Environment: Roads, other traffic, pedestrians, customers
- Actuators: Steering wheel, accelerator, brake, signal, horn
- Sensors: Cameras, sonar, speedometer, GPS, odometer, engine sensors, keyboard

# Ajan Programları - Örnek

- Agent: Medical diagnosis system
- Performance measure: Healthy patient, minimize costs, lawsuits
- Environment: Patient, hospital, staff
- Actuators: Screen display (questions, tests, diagnoses, treatments, referrals)
- Sensors: Keyboard (entry of symptoms, findings, patient's answers)

# PEAS Örnek: *Taksi süren ajan*

Başarım Ölçütü Performans Measure	Ortam Environment	Algılayıcılar Actuators	Tepki vericiler Sensors
Güvenlik, hızlı olmak, yasalara uymak, rahatlık, kazanç, doğru yere ulaşım	Yol, diğer araçlar, yayalar, yolcular	Kamera, sonar, hız ölçer, GPS, km sayacı, ivme ölçer, motor algılayıcıları, kontrol paneli	Direksiyon, gaz ve fren pedalı, korna, farlar, sinyaller vb.

- Şimdiki durumu bilmek için algılar kullanılır:
  - Araba şimdi nerede?
  - Arabanın şu anki hızı kaçtır?
  - Hareket koşulları nasıldır?

# PEAS Örnek-2

- Hastalıkların teşhisi için akıllı ajan
  - Başarıım ölçütü:  
Hastanın sağlığına kavuşması, masrafların ve yanlış teşhislerin minimum tutulması
  - Ortam:  
Hasta, hastane, hastane çalışanları
  - Tepki vericiler:  
Ekran, sorular, testler, tanılar, vb.
  - Alıcılar:  
Klavye, bulunan önceki vakalar, hastanın cevapları

# PEAS Example-2

- Intelligent agent for diagnosis of diseases

Performance criterion:

- Recovering the patient's health, keeping costs and misdiagnoses to a minimum

Environment:

- Patient, hospital, hospital staff

Actuators:

- The screen displays questions, tests, diagnoses, etc.

Sensors:

- Keyboard, previous cases found, patient's answers

# Ajan Türleri

- Basit tepki ajanı
- Model tabanlı ajan
- Hedef tabanlı ajan
- Fayda tabanlı ajan
- Öğrenme tabanlı ajan

# Types of Agents

Agents can be grouped into four classes based on their degree of perceived intelligence and capability:

- Simple Reflex Agents
- Model-Based Reflex Agents
- Goal-Based Agents
- Utility-Based Agents
- Learning Agent

# Basit Tepki (Refleks) Ajanları

- Basit bir koşul ve eylem sıralamasından ibaret olan vekiller
- Tanımlanan kural ve koşullara göre hareket ederler
- Belirlenen koşul gerçekleşince yine daha önceden belirlenen fiili yerine getirirler
  - Kurulu bir düzenek olarak düşünülebilir
- **Örneğin** fare kapanı, bir insan için fareyi yakalayan bir vekildir ve farenin peyniri yemesiyle birlikte fareyi yakalar
  - **Koşul:** peynir yenmesi
  - **Fiil:** farenin yakalanması

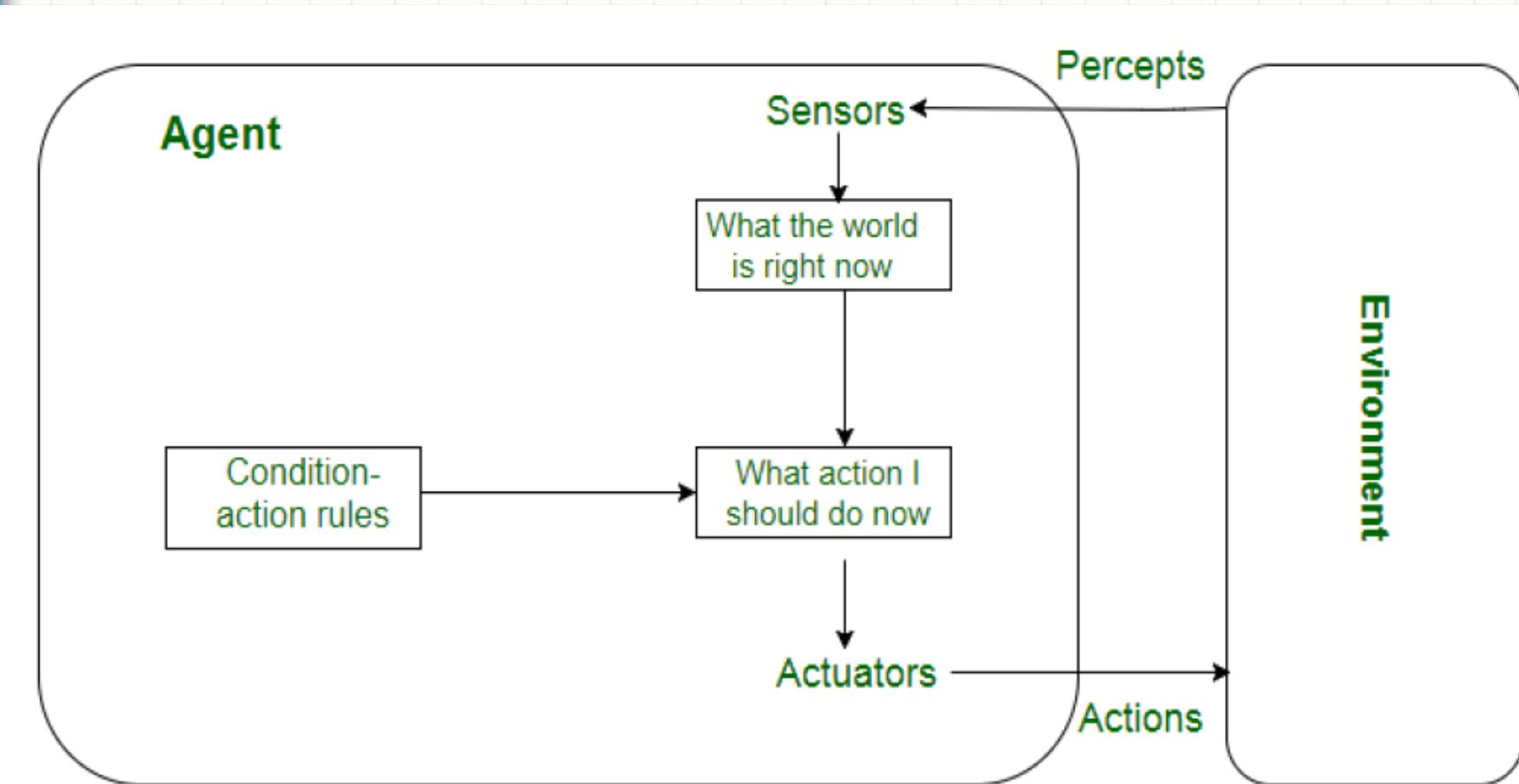
# Simple reflex agent

- Agents consisting of a simple sequence of conditions and actions
- They act according to the defined rules and conditions
- When the specified condition is fulfilled, they perform the predetermined action again.
  - It can be thought of as a set up
- For example, a mousetrap is a surrogate for a human to catch a mouse, and when the mouse eats the cheese, it catches the mouse.
  - Condition: eating cheese
  - Verb: catching a mouse

# Simple reflex agent

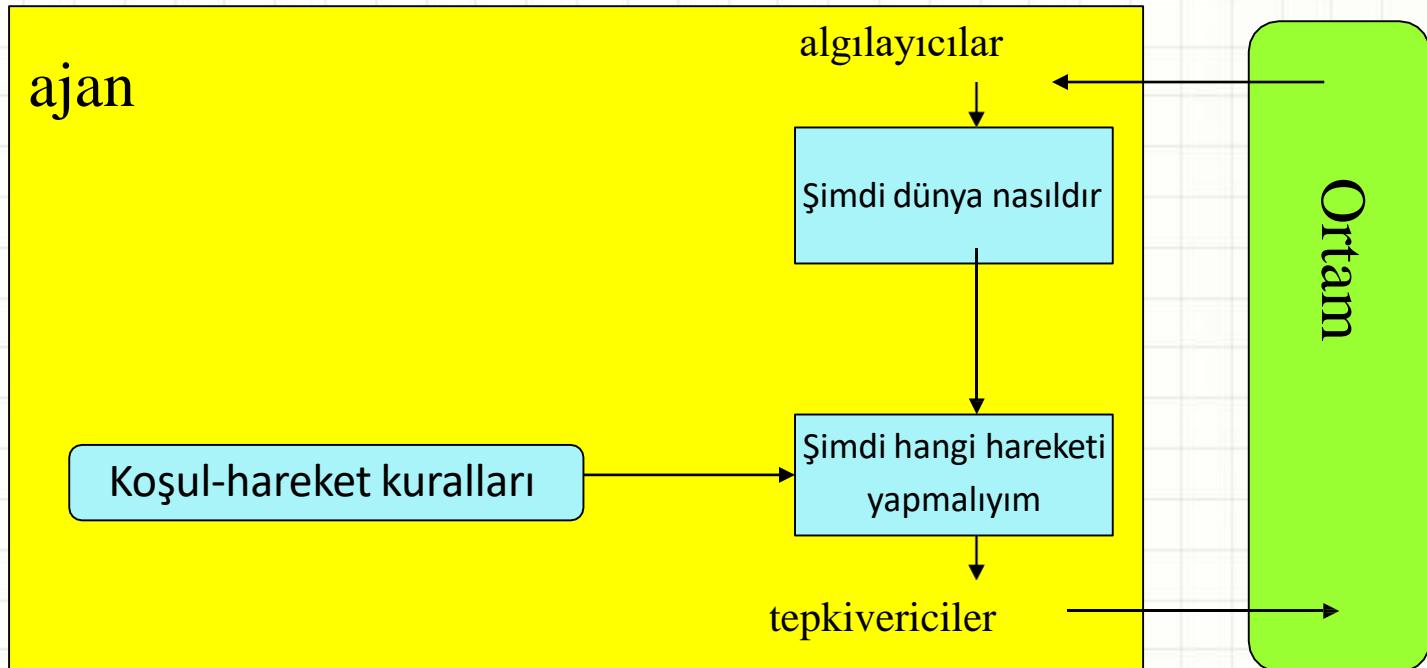
Problems with Simple reflex agents are :

- Very limited intelligence.
- No knowledge of non-perceptual parts of the state.
- Usually too big to generate and store.
- If there occurs any change in the environment, then the collection of rules need to be updated.



# Basit Tepki Ajanları

Ajanın hareketi yalnız şimdiki algılamalara bağlıdır, eski algılamalar dikkate alınmaz



- *Koşul-hareket kuralları*
- **Örnek: Eğer** öndeği araba frenlediyse, **o zaman** frenlemeyi başlat

# Basit Tepki Ajanları

- En basit ajan türü
- Geçmişteki duyumlarını yok sayarak, şu andaki duyuma göre karar verir
- Basit refleksli elektrikli süpürge ajanı için program:  
**function** REFLEKS-SÜPÜRGE-AJANI([konum,durum]) **returns** eylem  
    **if** durum = Kirli **then return** Temizle  
    **else if** konum = A **then return** Sağ  
    **else if** konum = B **then return** Sol
- Uygulama alanı **kısıtlıdır**

# Simple reflex agent

- The simplest type of agent
- Makes decisions based on current sensation, ignoring past sensations
- Program for simple reflex vacuum cleaner agent:
  - **function** *REFLEKS-SÜPÜRGE-AJANI([konum,durum]) returns eylem*
    - if** *durum = Kirli* **then return** *Temizle*
    - else if** *konum = A* **then return** *Sağ*
    - else if** *konum = B* **then return** *Sol*
- Application area is limited

# Model Tabanlı Ajanlar

- Dünyanın **şimdiki durumunun** yalnız o anki girişe değil bir önceki duruma da bakılarak saptanması gerekebilir
  - eski algılara bağlı olan durumların saklanması gereklidir
  - Buna **İç durum** (internal state) denir
- **Örneğin**, öndeeki araç frene bastı mı?
  - Fren lambalarını kontrol etmek gereklidir
  - Bu amaçla bir önceki görüntünün saklanması gereklidir
  - Önceki görüntüde lambalar sönük & şimdikinde yanıyor => frene **şimdi** basıldı

# Model-based agent

- The current state of the world may need to be determined by looking not only at the current entry but also at the previous state.
  - States that depend on old perceptions need to be kept
  - This is called the internal state.
- For example, did the vehicle in front apply the brakes?
  - Need to check brake lights
  - For this purpose, the previous image needs to be stored.
  - Lights off in previous image & on in current => brake is now applied

# Model Tabanlı Ajanlar

- Ajanın gözlemleri tam değilse (kısmen gözlemlenebilir dünya) kullanılır
- İç durum bilgisinin zamanla güncellenmesi, 2 tip bilginin ajan programında kodlanması gerektirir:
  - Dünyanın ajandan bağımsız olarak nasıl geliştiği hakkında bir miktar bilgi
  - Ajanın eylemlerinin dünyayı nasıl etkilediği hakkında bir miktar bilgi
- “Dünyanın nasıl işlediği” hakkındaki bu bilgi, dünyanın **modeli** olarak adlandırılır.

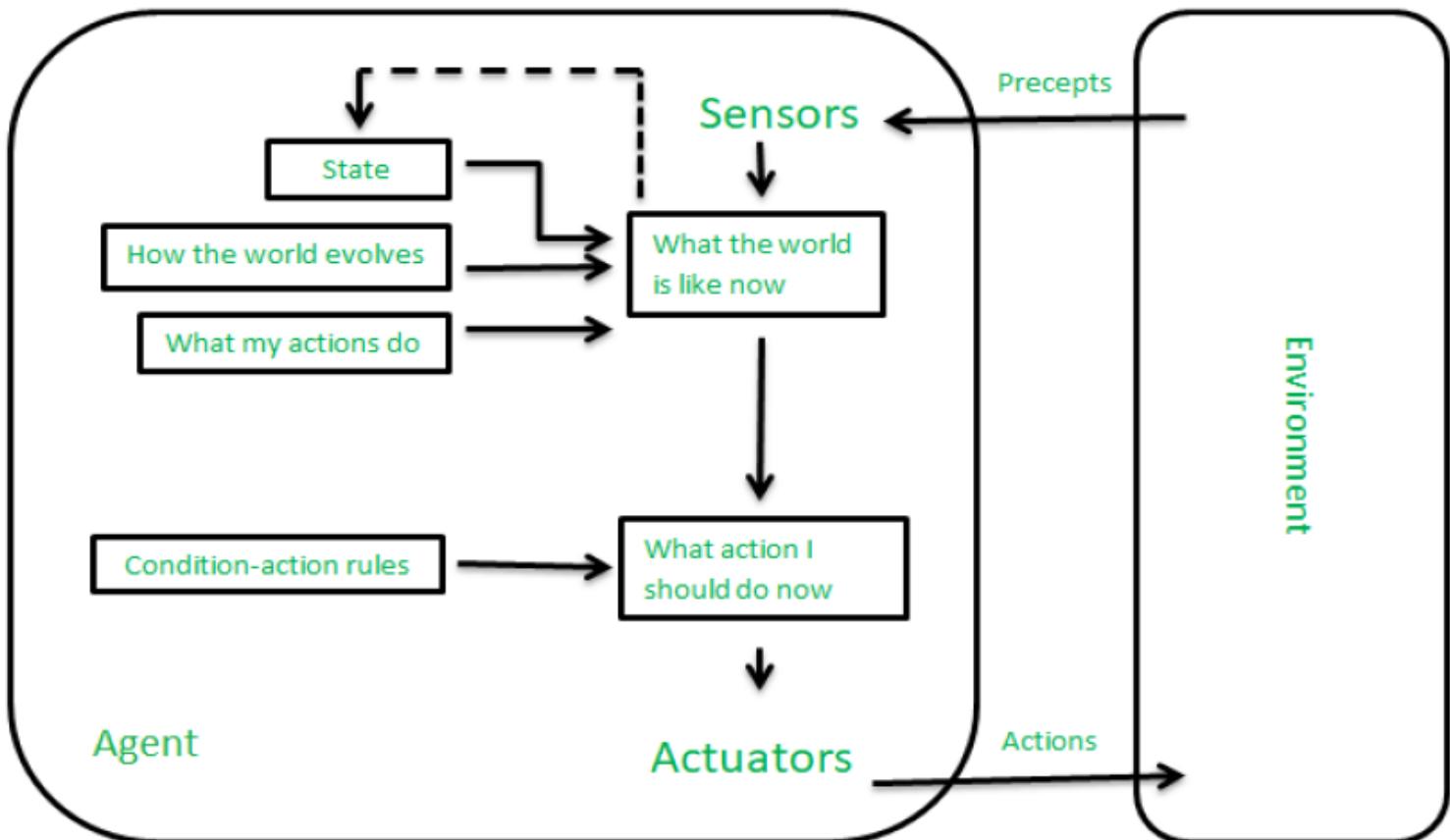
# Model-based agent

- Used if the agent's observations are incomplete (partially observable world)
- Updating internal state information over time requires encoding 2 types of information in the agent program:
  - A bit of information about how the world evolves independently of the agent
  - Some insight into how the agent's actions affect the world
- This knowledge of “how the world works” is called a model of the world.

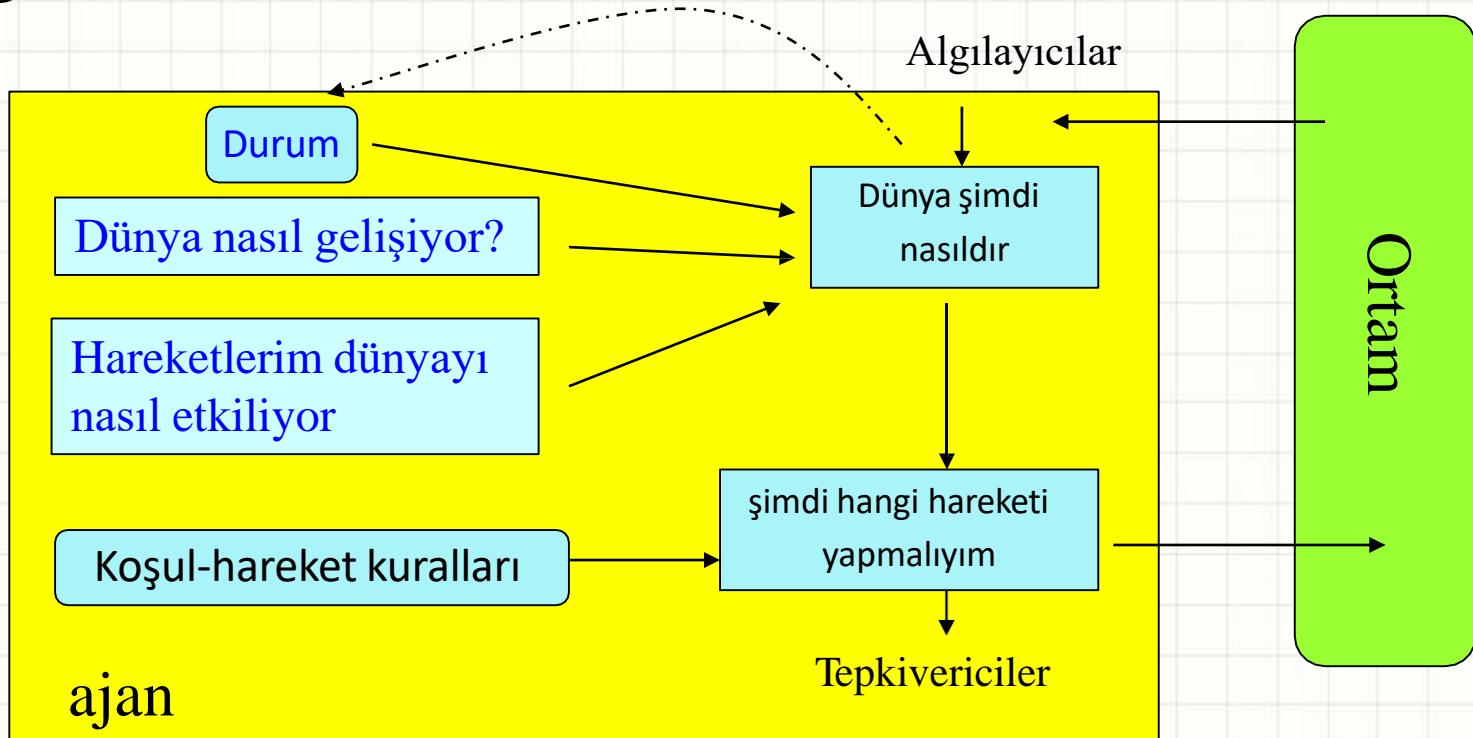
# Model-based agent

Updating the state requires information about :

- how the world evolves independently from the agent, and
- how the agent's actions affect the world.



# Model tabanlı ajanlar-Model-based agent



- Hareketi seçmek için İç durum bellekte tutuluyor
  - Algılayıcılar tüm dünya durumuna erişimi sağlayamıyor

# Model Tabanlı Ajanlar-Model-based agent

```
function Model_Tabanlı_Ajan(algı) returns eylem
    static: durum, /* mevcut dünyanın durumu */
             kurallar, /* koşul-eylem kuralları kümesi */
             eylem /* Son yapılan eylem */
    durum ← Durum_Güncelle(durum, eylem, algı)
    kural ← Kural_Karşılaştır(durum, kurallar)
    eylem ← Kural_Eylem[kural]
    durum ← Durum_Güncelle(durum, eylem)
    return eylem
```

- *Durum-güncellemeye*: yeni dünya durumları oluşturuluyor

# Hedef Tabanlı Ajanlar

- Ortamın şu andaki durumunu bilmek, ne yapılacağına karar vermek için her zaman yeterli değildir.
- Bazen ajan, şu andaki durum betimlemesi gibi, istenen durumları betimleyen bir tür **hedef** bilgisine ihtiyaç duyar
- **Arama** ve **planlama** Yapay Zeka'nın, ajanın hedeflerine ulaşmasını sağlayan eylem serilerini bulmaya adanmış alt alanlarıdır.
- Bu tip bir karar verme, daha önceden bahsedilen koşul-eylem kurallarından, **geleceği de hesaba kattığı için** temelde farklıdır.

# Hedef Tabanlı Ajanlar-Goal-based agent

- Knowing the current state of the environment is not always sufficient to decide what to do.
- Sometimes the agent needs some kind of goal-**target** information that describes desired states, such as the current state description.
- **Search and planning** are subfields of Artificial Intelligence dedicated to finding the action sequences that enable the agent to achieve its goals.
- This type of decision-making is fundamentally different from the condition-action rules mentioned earlier in that it also takes into account the future.

# Hedef Tabanlı Ajanlar

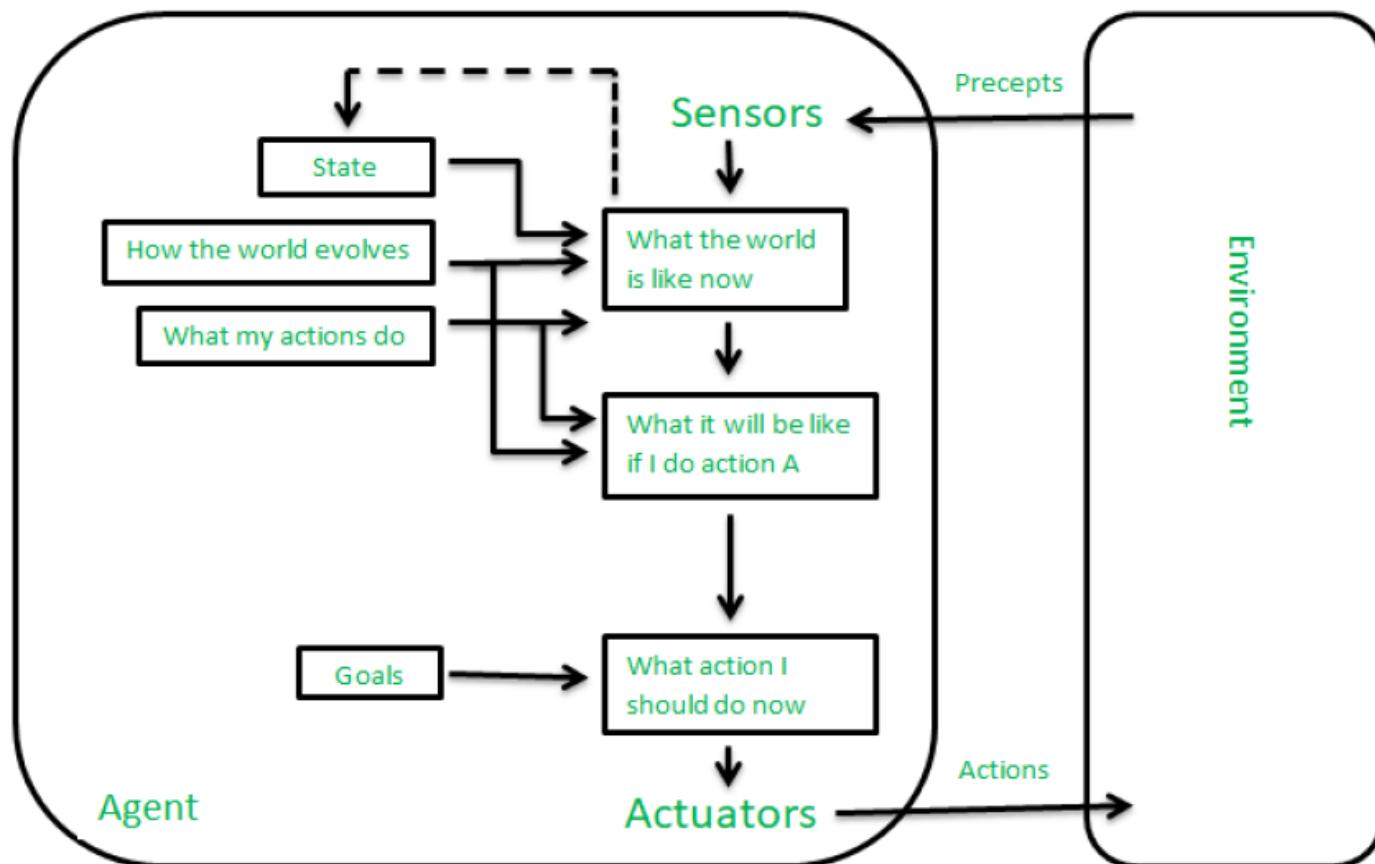
- **Basit Tepki Ajanı** fren lambasını gördüğü zaman fren yapar (**refleks olarak**)
- **Hedef Tabanlı Ajan** ise
  - Öndeki aracın fren lambaları yandığı zaman onun yavaşlayacağını **çıkarır**
  - Öndeki araca **çarpmama hedefini** gerçekleştirecek eylem ise fren yapmaktadır
- Her ne kadar hedef tabanlı ajan etkin görünmese de esnektir
  - **Örneğin** yağış başladığı zaman frenlerin etkin bir şekilde kullanılabilmesi için bilgisini yenileyebilir
  - **Basit Tepki Ajanı** için ise çok sayıda koşul-eylem kuralı yazmak gereklidir
  - **Hedef Tabanlı Ajanlarda** amacı değiştirerek farklı noktalara erişmek mümkündür. **Basit Tepki Ajanı** ise sadece bir noktaya gider

# Hedef Tabanlı Ajanlar- Goal-based agent

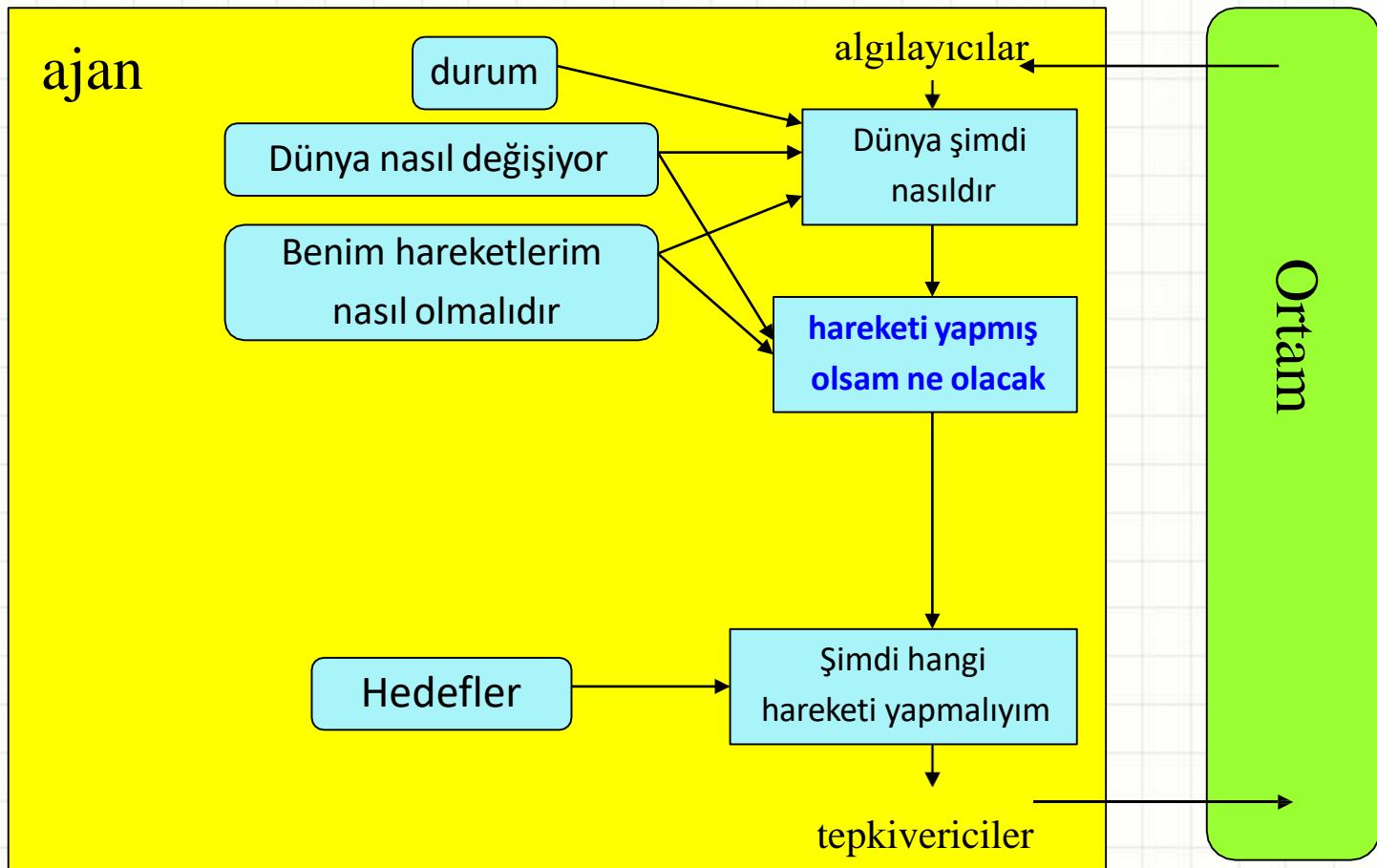
- The Simple Response Agent brakes (reflexively) when it sees the brake light
- If Target-Based Agent
  - ❖ Infers that when the brake lights of the vehicle in front turn on, it will slow down
  - ❖ Action to achieve the goal of not hitting the vehicle in front is to brake
- Although the target-based agent does not appear to be active, it is flexible
  - ✓ For example, when it starts to rain, the brakes are effectively can renew his knowledge so that it can be used
  - ✓ For Simple Response Agent, it is necessary to write many condition-action rules.
  - ✓ In Target Based Agents, it is possible to access different points by changing the purpose. Simple Response Agent only goes to one point

# Goal-based agent

These kinds of agents take decisions based on how far they are currently from their goal(description of desirable situations). Their every action is intended to reduce its distance from the goal. This allows the agent a way to choose among multiple possibilities, selecting the one which reaches a goal state. The knowledge that supports its decisions is represented explicitly and can be modified, which makes these agents more flexible. They usually require search and planning. The goal-based agent's behavior can easily be changed.



# Hedef Tabanlı Ajanlar- Goal-based agent



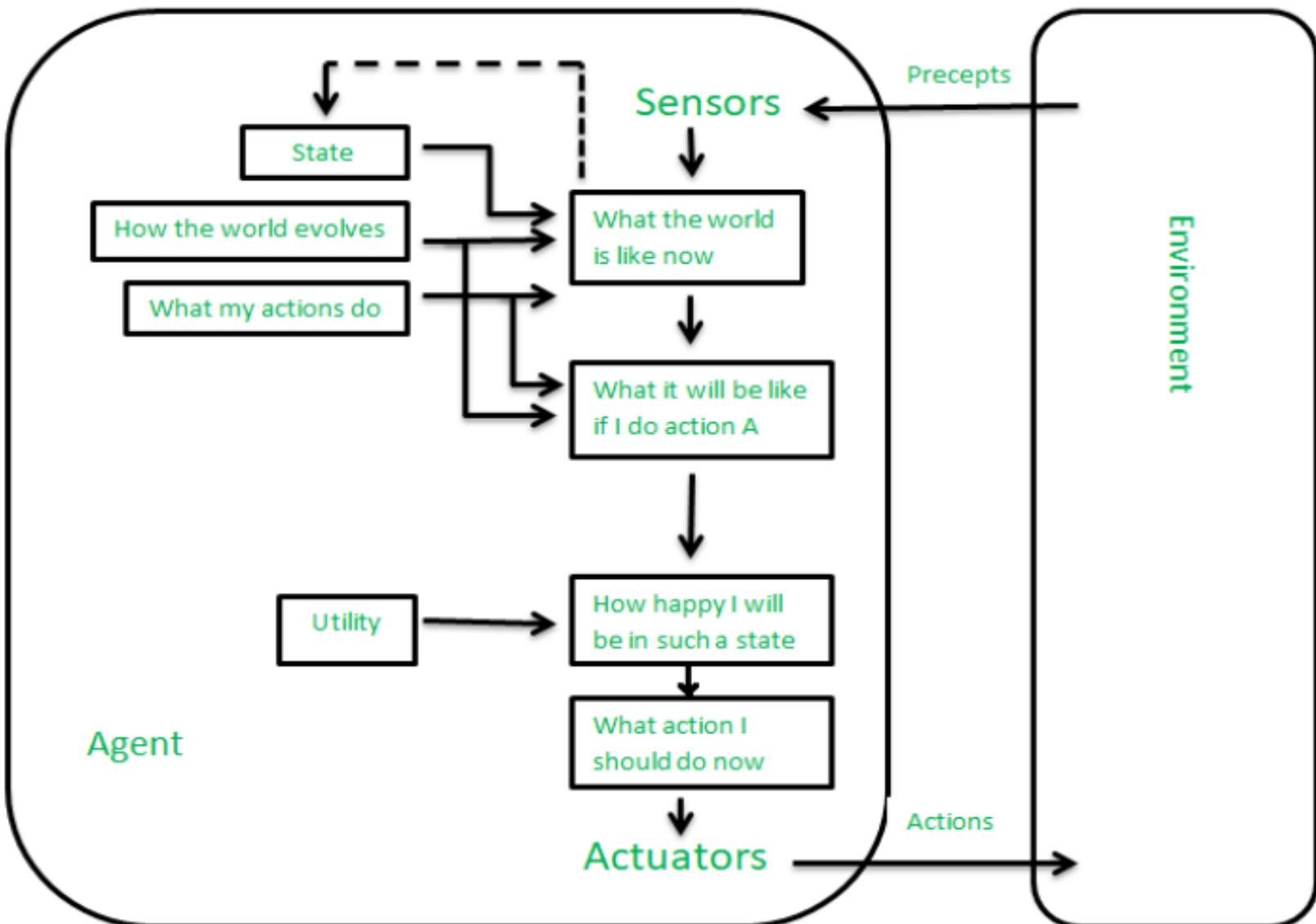
# Fayda Tabanlı Ajanlar- Utility-based agents

- Hedef bilgisinin yanında, eylemin ajanı ne ölçüde **mutlu** ettiği de dikkate alınır
- “**Mutlu**” terimi:
  - Bir dünya durumu diğerine tercih ediliyorsa, onun ajan için daha **faydalı** olduğunu ifade eder
- Daha çok tercih edilen durumun **fayda** değeri daha yüksektir
- Bir **fayda fonksiyonu** ile ölçülür
  - Herhangi bir durumu, tercih derecesine göre, sayısal bir değere dönüştüren fonksiyon

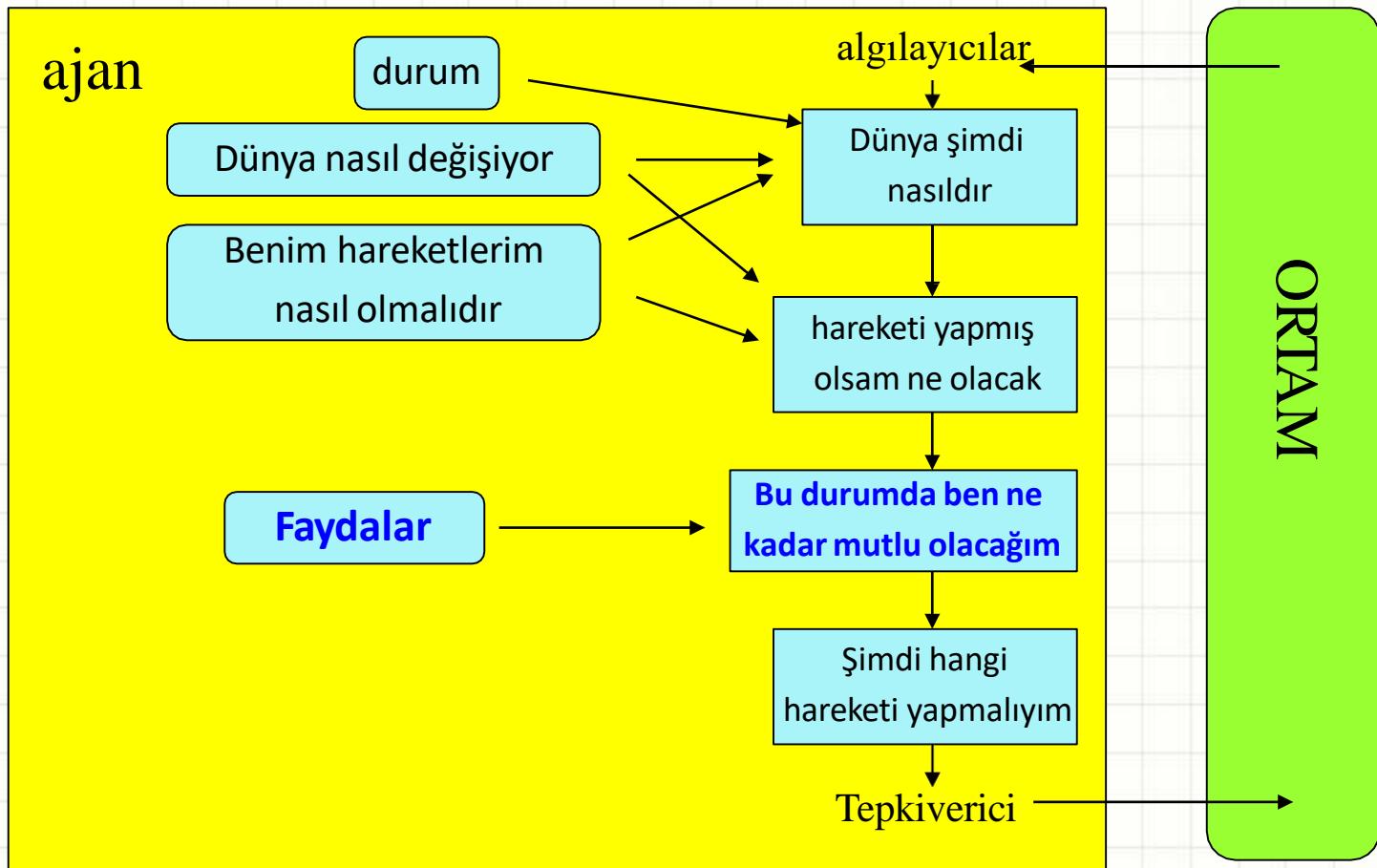
# Fayda Tabanlı Ajanlar: Utility-based agents

- To what extent does the agent of action, besides the target information, It is also taken into account that
- The term "happy":
  - It states that if one world state is preferable to another, it is more beneficial to the agent.
- The more preferred state has a higher utility value
- It is measured by a utility function
  - Function that converts any state to a numeric value, according to the degree of preference

# Utility-based agents



# Fayda Tabanlı Ajanlar



# Öğrenen Ajanlar

- Ortamda yapılan bazı eylemlerin beklenen sonuca nasıl hizmet ettiğine göre yeni kurallar tanımlanır
- Ajanın çalıştığı ortamın bilinmemesi halinde kullanışlıdırlar
- Kendi kurallarını ve durum makinelerini oluşturabilir veya değiştirebilirler
- Bu, ajanın başlangıçta bilinmeyen ortamlarda çalışabilmesini ve sadece başlangıç bilgisinin izin vereceği durumdan daha yetenekli hale gelmesini sağlar

# Öğrenen Ajanlar-Learning Agents

- New rules are defined according to how some actions in the environment serve the expected result.
- They are useful if the environment in which the agent is running is unknown
- They can create or modify their own rules and state machines
- This allows the agent to initially operate in unknown environments and become more capable than only initial knowledge would allow.

# Öğrenen Ajanlar

- Öğrenen bir ajan dört kavramsal bileşene sahiptir:
  1. Öğrenme Bileşeni
  2. Başarım Bileşeni
  3. Eleştirmen
  4. Problem Oluşturucu

# Öğrenen Ajanlar-Learning Agents

- A learning agent has four conceptual components:
  - 1.Learning Component
  - 2.Achievement Component
  - 3.Critic
  - 4.Problem Builder

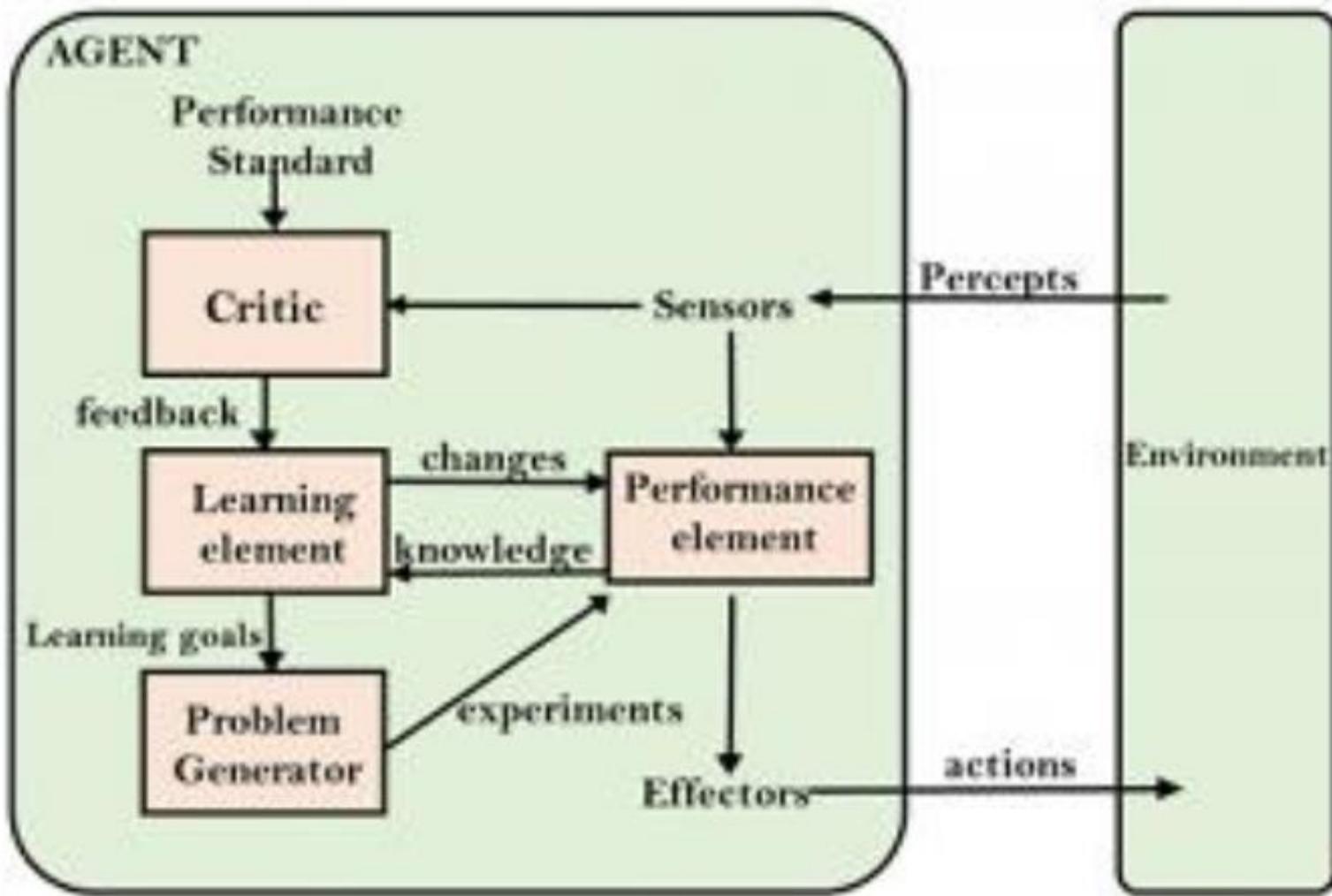
# Öğrenen Ajanlar

- **Başarım bileşeni**, duyumları (algıları) alır ve eylemlere karar verir
- **Öğrenme bileşeni, eleştirmenden** gelen ve ajanın ne kadar iyi olduğunu gösteren geri beslemeyi kullanarak, gelecekte daha iyi olması için başarım bileşeninin nasıl değiştirilmesi gerektigine karar verir.
- **Problem oluşturucu**, yeni ve bilgilendirici tecrübelere yol açacak eylemler önerilmesinden sorumludur

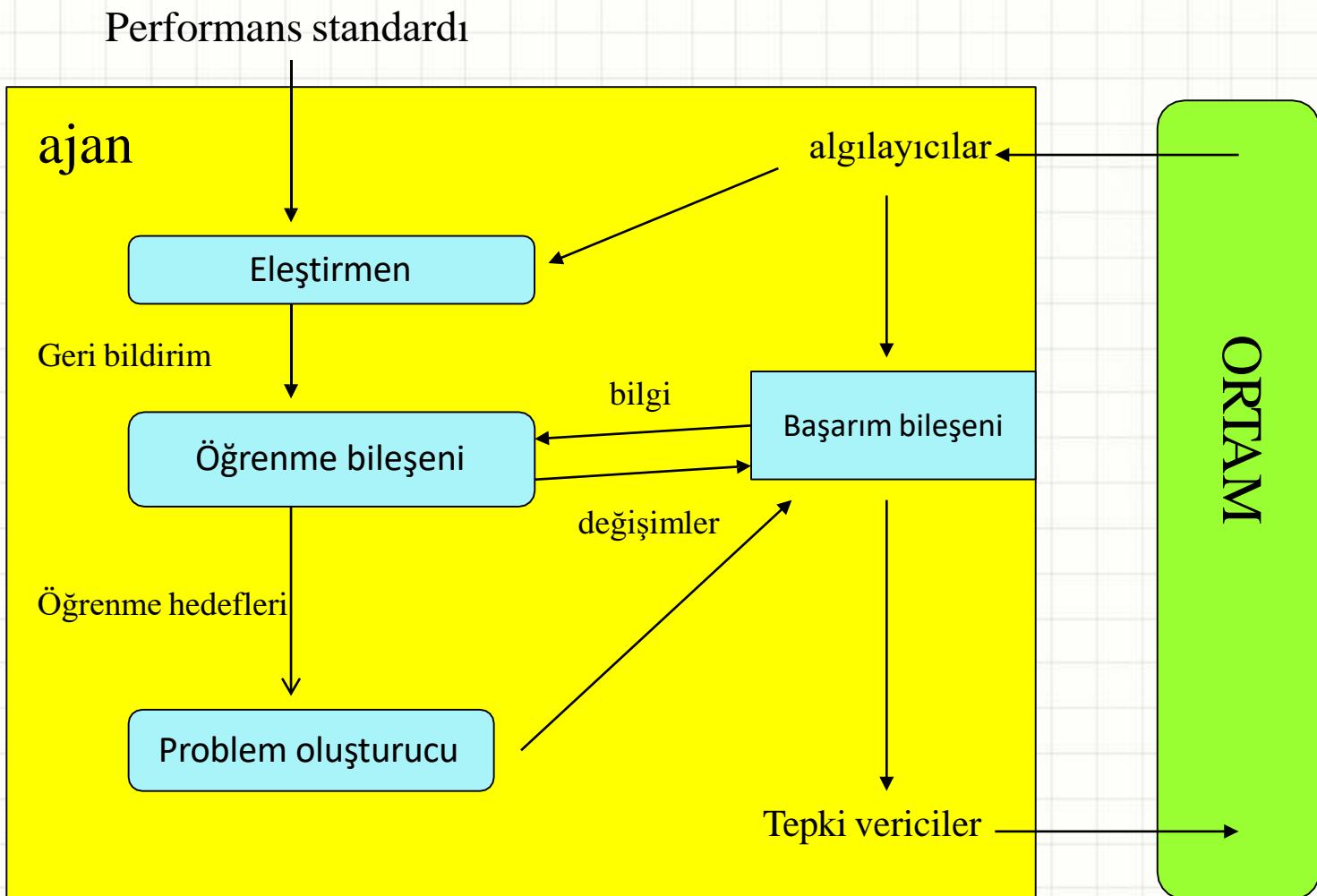
# Öğrenen Ajanlar-Learning Agents

- Performance component receives sensations (perceptions) and decides actions.
- Using feedback from the critic showing how good the agent is, the learning component decides how the performance component should be changed to be better in the future.
- The problem maker is responsible for suggesting actions that will lead to new and informative experiences.

# Learning Agents



# Öğrenen Ajanlar



# Özet - 1

- **Ajan**, bir ortamda algılayan ve eylemde bulunan bir şeydir.
- **Ajan fonksiyonu**, bir duyum serisine karşılık ajanın gerçekleştireceği eylemi belirler.
- **Performans ölçüyü**, ajanın ortamdaki davranışını değerlendirir.
- **Rasyonel ajan**, gördüğü duyum serisine karşılık, başarıım ölçütünün beklenen değerini en yüksek seviyeye çıkarmaya çalışır.
- **Görev ortamını**, başarıım, dış ortam, gerçekleştiriciler ve algılayıcıları içerir. Bir ajanı tasarlarken ilk adım mutlaka görev ortamının mümkün olduğunca tam olarak belirtilmesi olmalıdır.
- Görev ortamları, çeşitli boyutlar doğrultusunda değişim gösterir.
  - Bunlar tam veya kısmi gözlemlenebilir, deterministik veya stokastik, bağlantısız veya sıralı, durağan veya dinamik, ayrık veya sürekli ve tek ajanlı ya da çok ajanlı olabilir.

# Özet - Summary 1

- An agent is something that perceives and acts in an environment.
- The agent function is the agent's response to a series of sensations. determines the action to be taken.
- The performance measure evaluates the behavior of the agent in the environment.
- The rational agent tries to maximize the expected value of the performance criterion in response to the series of sensations it sees.
- It includes the task environment, performance, external environment, performers, and sensors. The first step in designing an agent should always be to specify the mission environment as precisely as possible.
- Mission environments vary along various dimensions.
  - They can be fully or partially observable, deterministic or stochastic, disconnected or sequential, static or dynamic, discrete or continuous, and single-agent or multi-agent.

# Özet - 2

- **Ajan programı**, ajan fonksiyonunu uygular. Açıga çıkarılan ve karar sürecinde kullanılan bilginin türünü yansitan çeşitli ajan programı tasarımları vardır.
- **Basit tepki ajanları** doğrudan duyuma yanıt verir.
- **Model tabanlı ajanlar**, dünyanın o andaki duyumda açıga çıkmayan açılarını takip etmek için içsel durumlarını kullanır.
- **Hedef tabanlı ajanlar**, hedeflerine ulaşmak için eylemde bulunur.
- **Fayda tabanlı ajanlar** beklenen “mutluluk” (**fayda**) değerini en yüksek seviyeye çıkarmaya çalışır.
- Bütün ajanlar başarılarını **öğrenme** yoluyla arttıracaktır.

# Özet – Summary 2

- The agent program implements the agent function. There are various agent program designs that reflect the type of information that is disclosed and used in the decision process.
- Simple response agents respond directly to sensation.
- Model-based agents use their internal states to follow aspects of the world that are not revealed in the current sense.
- Goal-based agents take action to achieve their goals is found.
- Utility-based agents seek to maximize the expected value of "happiness" (utility).
- All agents can increase their performance through learning.