



# »Лекционен курс



# »Интелигентни системи



## Агенти със знания»<sup>1</sup>

# Увод

## » Агентите със знания могат да:

- > Правят **комплексни представяния** (знания) на света
- > Извличат нови представяния (знания) посредством **процес на извод**
- > Основавайки се на тези нови представяния (знания) да **правят заключения** за това, какво трябва да се прави



# База знания

» Основен компонент на тези агенти е База знания (БЗ)

- > Състои се от множество от съждения
- > Всяко съждение е представено в някакъв език за представяне на знания
- > Използват се също „аксиоми“
  - + Съжденията, приети като дадени, без да са изведени от други съждения



# База знания

- » Трябва да има възможност за:
  - > Добавяне на нови съждения (Tell)
  - > Правене на запитвания (заявки) (Ask)
- » Двете действия могат да съставляват правене на изводи (inference)
  - > Т.е. извод на нови съждения от стари съждения
- » Един извод трябва да удовлетворява следното изискване:
  - > Когато се пита една БЗ (Ask), отговорът трябва да следва от това, което е въведено преди това (Tell)



# Генетичен агент със знания

```
function KB-Agent (percept) returns едно действие
  persistent: KB, база знания
               t, брояч, който дава времето, първоначално 0

  Tell(KB, Make-Percept-Sentence(percept, t));
  action  $\leftarrow$  Ask(KB, Make-Action-Query(t));
  Tell(KB, Make-Action-Sentence(action, t));
  t  $\leftarrow$  t + 1;
  return action
```

# Коментар

## » Както всички агенти:

- > Като вход получава възприятията на агента
- > Като изход връща едно действие

## » Поддържа БЗ, която може да съдържа някакви предварителни фонові знания

## » Програмата прави следните неща:

- > Записва (Tell) в БЗ какво възприема
- > Пита (Ask) БЗ кое действие да извърши
  - + При това може да се наложи правене на изводи върху актуалното състояние на света
- > Споделя (Tell) на БЗ избраното действие и оценява това действие





# Опериране на генетичен АЗ

```
function KB-Agent (percept) returns едно действие
  persistent: KB, база знания
               t, брояч, който дава времето, първоначално 0

  Tell(KB, Make-Percept-Sentence(percept, t));
  action ← Ask(KB, Make-Action-Query(t));
  Tell(KB, Make-Action-Sentence(action, t));
  t ← t + 1;
  return action
```

# Език за представяне

- » Детайлите на езика за представяне са скрити в три функции, които реализират:
  - > Интерфейсът между сензорите и ефекторите от една страна и
  - > Базовото представяне и системата за заключения – от друга





# Функции

## » Make-Percept-Sentence

- > Конструира съждение, приемайки че агентът е приел даденото възприятие в определеното време

## » Make-Action-Query

- > Конструира съждение, което пита кое действие трябва да изпълни в актуалното време

## » Make-Action-Sentence

- > Конструира съждение, което осигурява избраното действие да бъде изпълнено

## » Детайлите на **машината за извод** са скрити в Tell и Ask



# Обобщение

- » Един АЗ е аналогичен на **агент с вътрешни състояния**
  - > основавайки се на Tell и Ask не е произволна програма за изчисляване на действия
- » Той е сравним с едно описание на ниво знания, където, за да определим неговото поведение трябва да специфицираме:
  - > Какво **знае** агентът и
  - > Какви **цели** има



# Пример

## » Автоматизирано такси

- > Има за **цел** един студент да бъде закаран от нова сграда до ректората на ПУ
- > **Знае**, че се намира в Пловдив и мостовете на Марица са единствената връзка между двете локации

» Извод: Можем да очакваме, че **таксито ще премине през някои от тези мостове**, понеже трябва да постигне целта си



# Пример

- » Забележете, че този анализ е **независим** от това как оперира таксито на имплементационно ниво
- » Няма значение:
  - > Как са представени географските данни – като списък от символи или пиксели
  - > Прави изводите като манипулира символи или по друг начин
  - > Предава шумовите сигнали в невронна мрежа ...

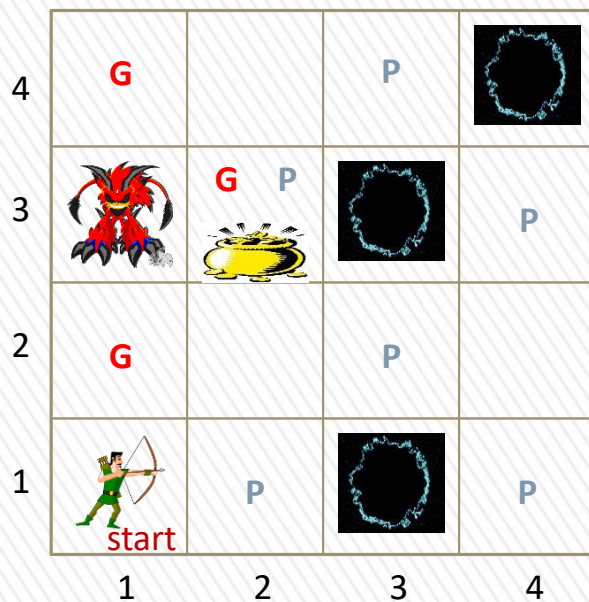


# Декларативен и процедурен подход

- » Изводът беше направен като използвахме един **декларативен подход**
  - > Противно на **процедурния подход**, който кодира желаното поведение **директно като програмен код**
- » През 70-те и 80-те години на миналия век – горещи спорове между поддръжниците на двата подхода
  - > Днес, разбираме, че един успешен агент трябва да може **да комбинира двата подхода**
  - > Също, декларативните знания в много случаи могат да бъдат **компилирани** в ефективен процедурен код
- » Един АЗ може да бъде снабден също с **механизъм за учене**
  - > Такъв АЗ може да бъде **напълно автономен**



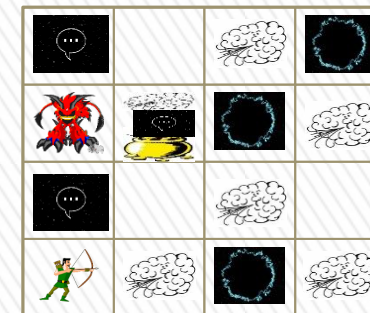
# Въвеждащ пример



Ще разгледаме една среда, позната като Wumpus (W.), в която АЗ може да демонстрира възможностите си



# Въвеждащ пример



PEAS (Performance, Environment, Actuators, Sensors):

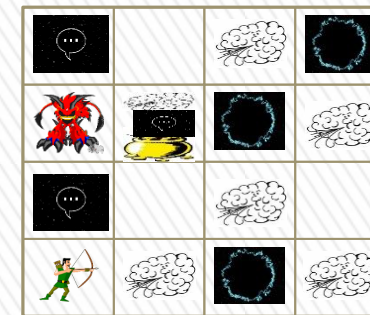
**Performance:**

- +1000 – излиза от пещерата със златото
- -1000 – пада в дупка или изяден от чудовището
- -1 – всяко предприето действие
- -10 – за използване на единствената стрела

**Environment:** 4 X 4 матрица

- Започва пътя винаги в полето [1,1], като гледа надясно
- W. и златото – местоположението, определено на основата на случайно равномерно разпределение, но не в началното поле
- Всяко поле (без началното) съдържа дупка с вероятност 0.2

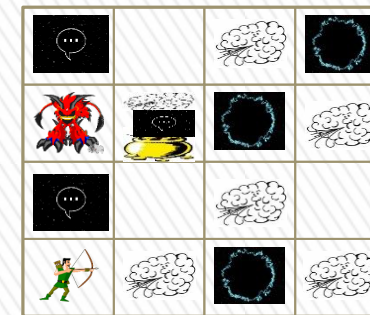
# Въвеждащ пример



## Effectors:

- Може да се движи ( *Forward, Turnleft, Turnright* )
- Движение напред няма ефект, когато стои пред стена
- При среща с W. или падане в дупка – ужасна смърт
- Действие Grab – взема предмет, който се намира в същото поле
- Действие Shoot – изстрелва стрелата в посоката, в която е агентът. Стрелата лети, докато срещне W. или стена. Има само една стрела, т.е. само първият Shoot има ефект
- Действие Climb – излиза от пещерата, само от полето [1,1]

# Въвеждащ пример

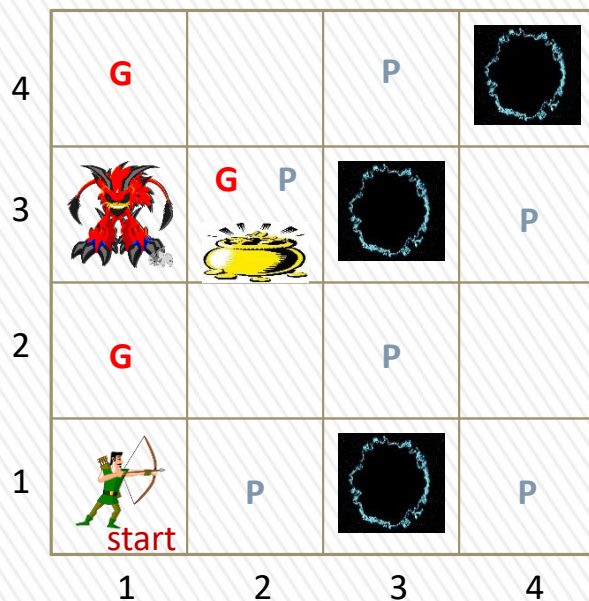


**Sensors:** снабден с 5 сензора, които доставят различна информация

- Stench – във всяко поле, където се намира W. , и съседните (не по диагонал) се носи зловоние
- Breeze – течение в полетата, съседни на дупка (не по диагонал)
- Glitter – полето със златото
- Bump – агентът се удря в стена
- Scream – рев, когато W. е улучен, чува се във всички полета

Възприятия: списък с 5 символа, напр.  
[Stench, Breeze, None, None, None]

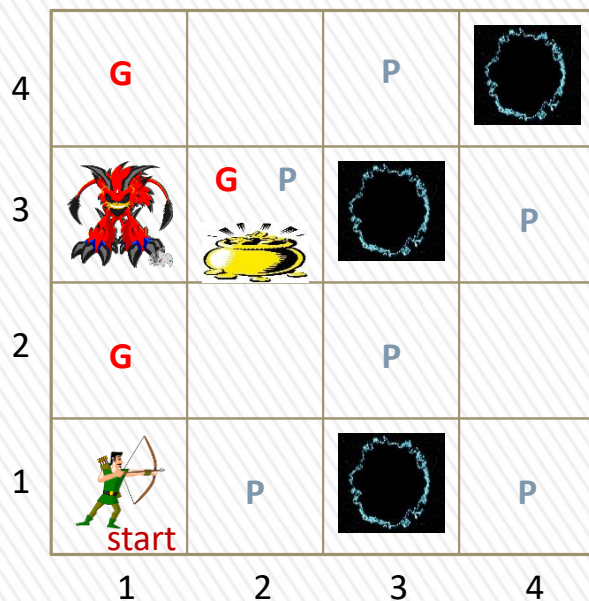
# Пример



- Дискретна, статична среда ( $W$ . неподвижен)
- Частично наблюдаема: някои аспекти на състоянието не могат да бъдат наблюдавани:
  - Позиция агент
  - Здравно състояние на  $W$ .
  - Наличие на стрела
- Позициите на ями и  $W$ . – приемаме ги като ненаблюдаеми елементи на състоянието
- Моделът на преход за средата е напълно познат

- АЗ работи последователно, т.е. наградата след като се извършат поредица от действия
- Моделът на преход за агента е непознат: не знае кое действие е фатално
  - Знанията на агента за модела на преход се попълват чрез разкриване на полетата с ями и  $W$ .

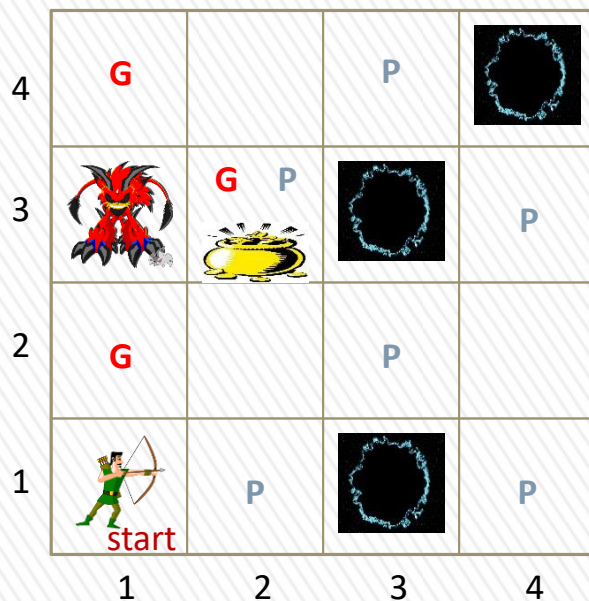
# Пример



- АЗ работи последователно, т.е. наградата след като се извършат поредица от действия
- Моделът на преход за агента е непознат: не знае кое действие е фатално
  - Знанията на агента за модела на преход се попълват чрез разкриване на полетата с ями и W.



# Пример



- Най-голямата трудност за АЗ е неговото начално незнание за конфигурацията на средата
- За преодоляване на това незнание ще бъдат необходими логически заключения
- За повечето инстанции на W. света, АЗ може да намери златото без наранявания
- В други случаи агентът трябва да решава, дали да остане с празни ръце или да намери златото с риск от смърт
- Около 21% средите са напълно некоректни – златото се намира над яма или е заградено от ями
- Ще разгледаме как един АЗ **разучава средата** – използваме **език за представяне на знания като растер**, върху който се задават символи



# Пример

Възприятия: начално състояние  
[None, None, None, None, None]

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2	3,2	4,2
1,1 A OK	2,1 OK	3,1	4,1

A	Агент
B	Полъх (Breeze)
G	Блести, злато (Glitter, Gold)
OK	Сигурно поле
P	Яма (Pit)
S	Зловоние (Stench)
V	Посетен (Visited)
W	Wumpus

БЗ на агента съдържа първоначално правилата на средата, както бяха представени

Освен това знае, че се намира в поле [1,1] и полето е сигурно

На основата на възприятието агентът заключава, че съседните полета [1,2] и [2,1] могат да бъдат безопасно посетени

## пример

Възприятия: след първия ход  
[None, Breeze, None, None, None]

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 OK	2,2 P?	3,2	4,2
1,1 V OK	2,1 B A OK	3,1 P?	4,1

A	Агент
B	Полъх (Breeze)
G	Блести, злато (Glitter, Gold)
OK	Сигурно поле
P	Яма (Pit)
S	Зловоние (Stench)
V	Посетен (Visited)
W	Wumpus

Един внимателен агент посещава едно поле само ако знае, че е OK

Приемаме, че агентът решава да посети полето [2,1]

В това поле възприема „полъх“ – означава, че в някои от съседните полета има яма

В [1,1] няма, тогава трябва да бъде в [2,2] или [3,1]

# Пример

Възприятия: след първия ход  
[None, Breeze, None, None, None]

1,4	2,4	3,4	4,4
1,3	2,3	3,3	4,3
1,2 A OK	2,2 P?	3,2	4,2
1,1 V OK	2,1 B A OK	3,1 P?	4,1

A	Агент
B	Полъх (Breeze)
G	Блести, злато (Glitter, Gold)
OK	Сигурно поле
P	Яма (Pit)
S	Зловоние (Stench)
V	Посетен (Visited)
W	Wumpus

В този момент агентът знае, че само едно поле е сигурно – агентът се обръща и се връща в [1,1] и продължава в [1,2]

# Пример

Възприятия: след третия ход  
[Stench, None, None, None, None]

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 P! OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

Това е сравнително трудно заключение, понеже комбинира знания, които са получени в **различни времена, на различни места**

A	Агент
B	Полъх (Breeze)
G	Блести, злато (Glitter, Gold)
OK	Сигурно поле
P	Яма (Pit)
S	Зловоние (Stench)
V	Посетен (Visited)
W	Wumpus

В [1,2] агентът възприема зловоние

Това означава, че някъде наоколо се намира W. – не е в [1,1] и [2,2] (иначе, в [2,1] “S”)

Така агентът заключава, че W. се намира в [3,1]

Понеже преди това е заключено, че в [2,2] или [3,1] трябва да има яма, тук можем да актуализираме знанието на агента, че тя е в [3,1]



# Пример

Възприятия: след петия ход  
[Stench, Breeze, Glitter, None, None]

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 S A B G	3,3 P?	4,3
1,2 S V OK	2,2 V A OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

A	Агент
B	Полъх (Breeze)
G	Блести, злато (Glitter, Gold)
OK	Сигурно поле
P	Яма (Pit)
S	Зловоние (Stench)
V	Посетен (Visited)
W	Wumpus

Агентът се е убедил, че в [2,2]  
няма нито яма, нито W., т.е. то е  
OK и може да го посети

От тук лесно намира златото



# Пример

Възприятия: след петия ход  
[Stench, Breeze, Glitter, None, None]

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 S A B G	3,3 P?	4,3
1,2 S V OK	2,2 V A OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

**Обобщение:** когато агентът прави заключения от наличната информация, тези заключения са **гарантирано коректни**, когато наличната информация е **коректна** – това е **основополагащо свойство на логическите заключения**





Благодаря за вниманието!