

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Претставување на знаењето

# Содржина на лекцијата

- Претставување на знаењето
- Репрезентациски шеми
  - ☐ Логички
  - ☐ Процедурални
  - ☐ Мрежни
  - ☐ Структурни
- Проблеми при претставувањето на знаењето

# Претставување (репрезентација) на знаењето

- Претставувањето на знаењето е еден од централните проблеми во вештачката интелигенција.
- $\pi = 3,141592\dots$
- Една од суштините на ВИ е да се најдат соодветни претставувања на знаењето со кои полесно ќе се реши проблемот.

# Уште неколку прашања:

- Дали се согласувате со централната догма на вештачката интелигенција, според која сè се базира на пресметување?
- Дали постои начин да се претстави знаењето?
- Колку треба да биде голем еден комплетен систем за претставување на знаењето?
- Какво претставување на знаењето би користел тој систем?

# Расудување со знаењето

- Кога луѓето размислуваат, не ги следат секогаш правилата за дедукција.
- Сакаме да веруваме сека живееме во свет во кој постои ред и тој е предвидлив.
- Се користат различни стратегии и тактики:
  - Генерализација (индуктивно расудување)
  - Објаснување
  - Дедукција
  - Визуелизација ...

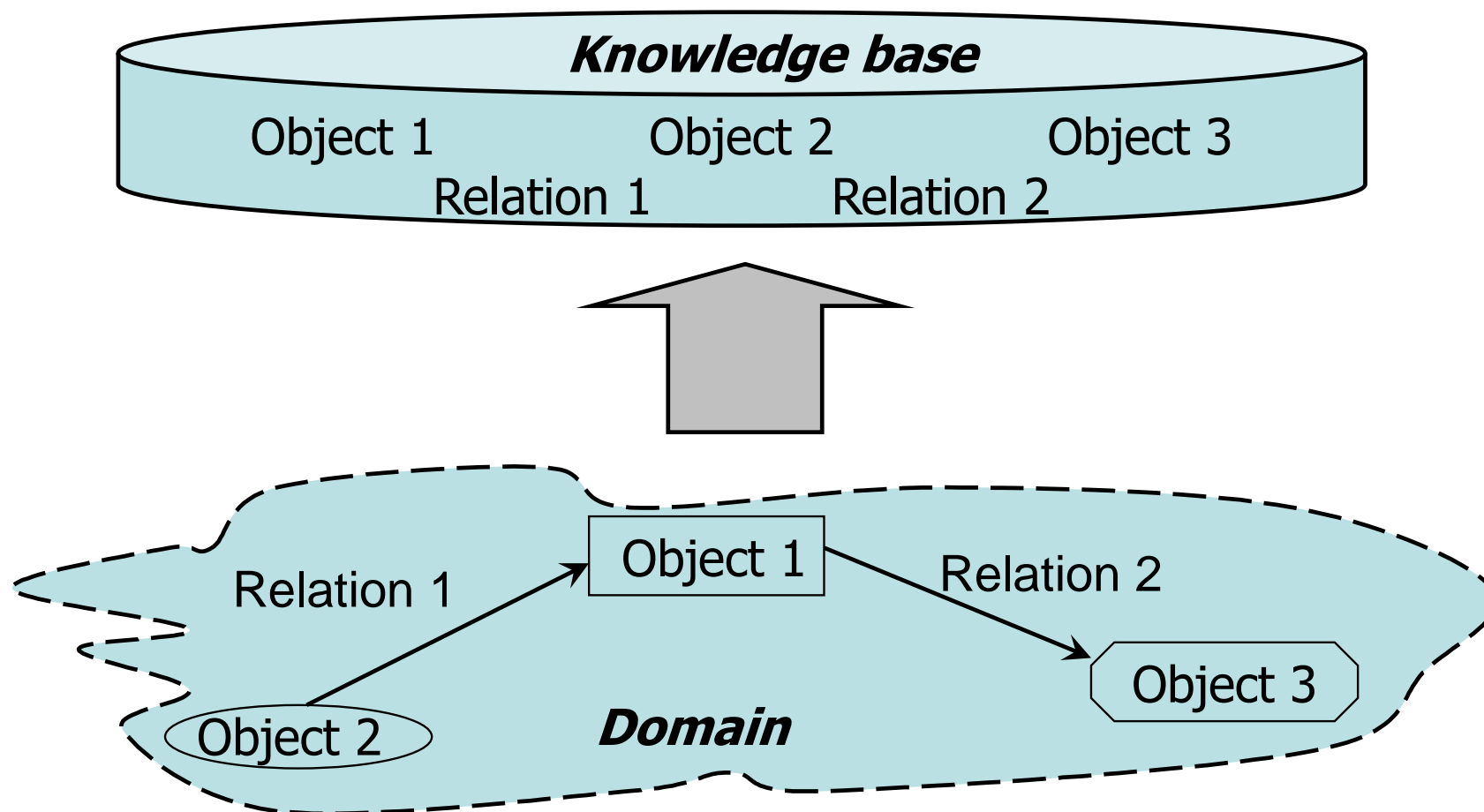
# Претставување на знаењето

- При градењето на база на знаење инженерот мора да ги избере значајните објекти и релации од доменот на проблемот и да ги преслика во некој формален јазик.
- Програмата треба да има доволно „знаење“ (факти и правила) за да може да решава проблеми од доменот, да изведува правилни заклучоци од ова знаење и сето ова да го прави ефикасно.

# Претставување на знаењето

- Резултатот на изведувањето од базата на знаење треба да соодветствува на резултати од акции или појави во реалниот свет.
- Објектите, релациите и изведувањето што се расположливи на програмерот зависат од изборот на јазикот за опис на знаење.
- Соодветната претстава може да ги олесни прибирањето (аквизицијата), организацијата и отстранувањето на грешки во базата на знаење.

# Претставување на знаењето



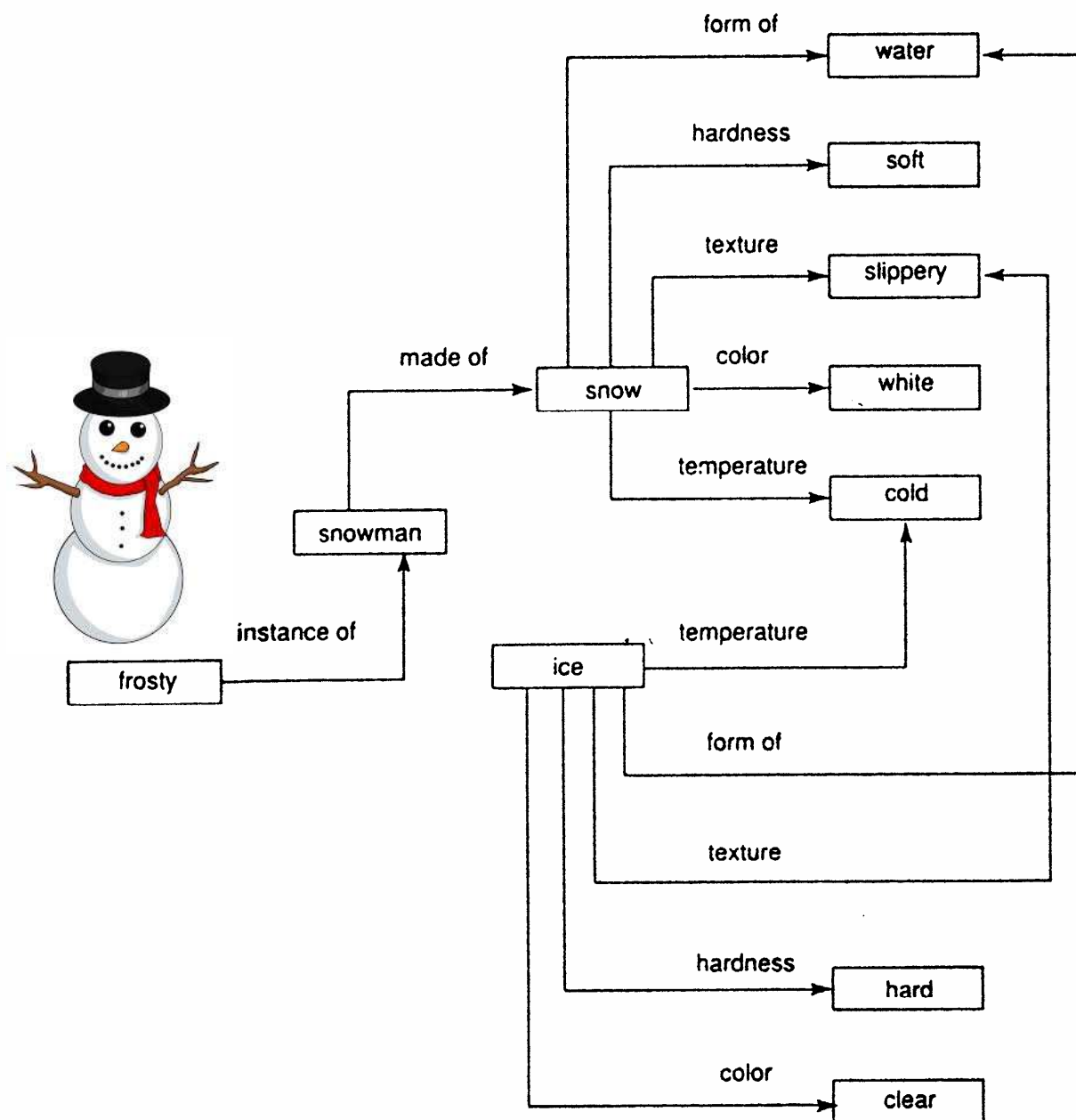


# Концепти (Поими)

- Кога луѓето го спознаваат некој објект и резонираат околу него, ова спознание се формира во концепт (поим)
- Концептот е дел на нашето севкупно знаење и е поврзан со соодветни релации со други концепти.
- На пр. Од искуство, концептот за *снег* го поврзуваме со концептите како: *студено, бело, снешко, лизгаво, мраз.*

# Претставување на концептите

- Не е доволно да се знае дека фактот „Снегот е бел“ е вистина.
- Потребна е претстава која би можела да одговори на прашањата како што се:
  - „Од што е составен снегот?“
  - „Која е температурата на снешкото Мразулко?“
  - „Ако снегот е бел, тогаш која боја е Мразулко?“

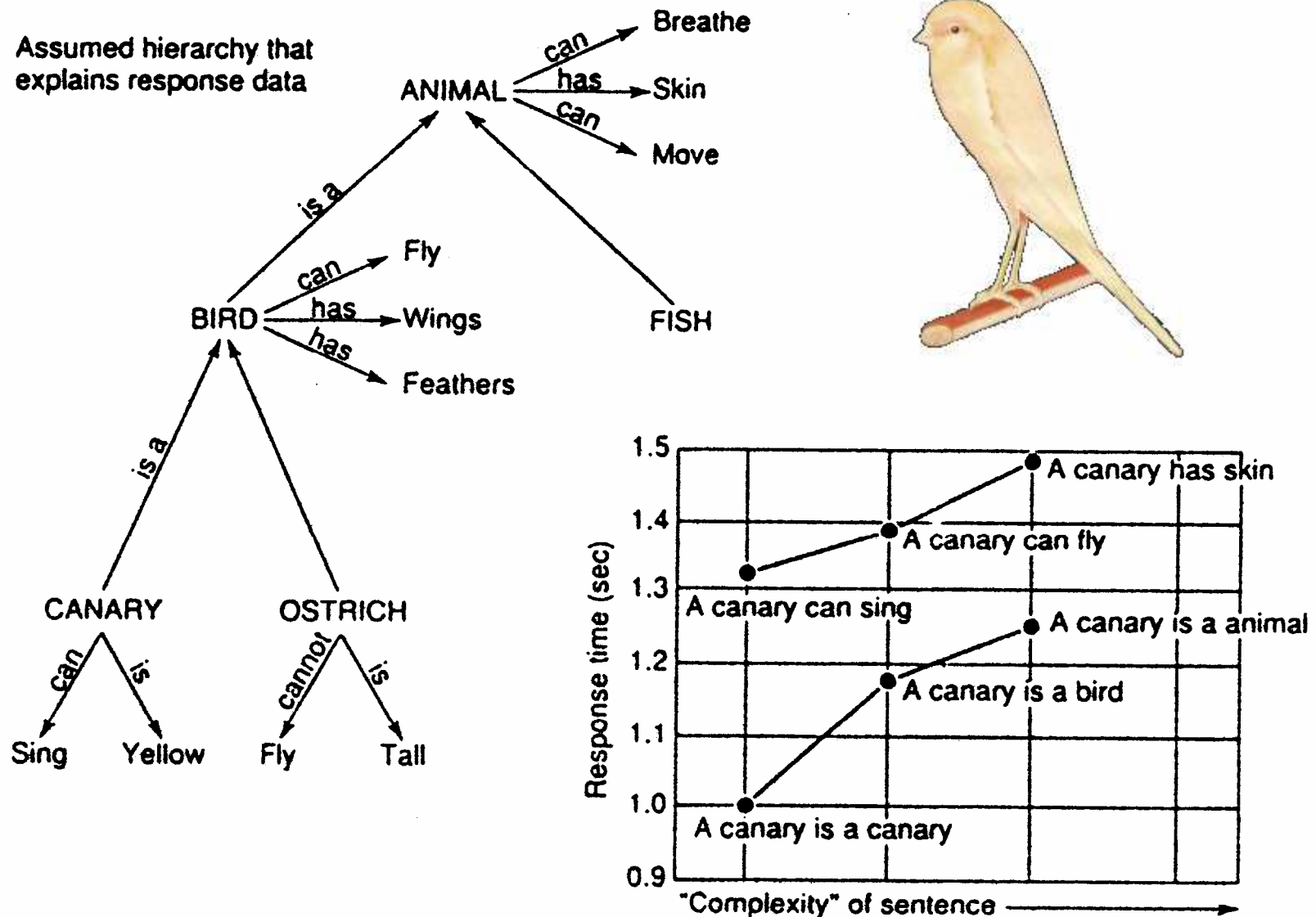


# Општи принципи

- Општи принципи при организација на знаење на пр. хиерархија на класи
- Како?
  - ☐ Општ механизам за нивна претстава (претставување, репрезентација)
  - ☐ претставување на дефиниции
  - ☐ правила и исклучоци
  - ☐ информации кои недостасуваат и подразбирани (предодредени) вредности
  - ☐ претстава на време
  - ☐ причинско - последични односи
  - ☐ несигурност

# Хиерархиска организација на ЧОВЕКОВОТО ЗНАЕЊЕ

- Постојат психо-физички експериментални докази кои наведуваат на заклучокот дека човековото знаење е хиерархиски организирано. Мерено е времето потребно испитаниците да одговорат (response time) на прашања како што се:
  - ☐ „Дали канаринецот е птица?“,
  - ☐ „Дали канаринецот пее?“
  - ☐ „Дали канаринецот лета?“



**Figure 9.1** Semantic network developed by Collins and Quillian in their research on human information storage and response times (Harmon 1985).

# Хиерархиска репрезентација на знаењето

- Придобивки од хиерархиската претстава на знаење (наследување)
  - се избегнува непотребното дуплирање на факти заеднички за поголема група
  - Лесно одржување на конзистентна база на знаење
  - Лесно додавање нови класи и индивидуи

# Шема и медиум за претставување

- Разграничување помеѓу шемата и медиумот за репрезентација. Слично на податочни структури (медиум) и програмски јазици (шема) за имплементација.
- Јазиците за претставување на знаење вообичаено се поограничени од предикатното сметање и програмските јазици



# Репрезентациски шеми

- Во изминатите 45 години предложени се голем број репрезентациски шеми (секоја со свои предности и слабости) кои може да се поделат на 4 главни категории:

## 1. Логички

- предикатно сметање
- исказна логика

## 2. Мрежни

- семантички мрежи
- семантички молекули
- концептуална теорија на зависност
- онтологии

## 3. Процедурални

- IF...THEN... правила

## 4. Структурни

- скрипти
- рамки

# 1. Логички репрезентациски шеми

- Оваа класа на репрезентации користи изрази во формална логика, правила за изведување и процедури за докажување
- Предикатно сметање од прв ред
- PROLOG

# Формално расудување (formal reasoning)

- Логичко расудување
- Услов: Исказите кои се дадени *декларативно* (пр. “Децата имаат весел поглед.”) треба да се претстават *условно* (“Додека си дете, погледот ти е весел.” или “Кога си дете, тогаш погледот ти е весел.”, односно на крај “Ако си дете, тогаш погледот ти е весел”).

# Начин на претставување на знаењето

- Експлицитен избор на референците, т.е сето она што постои, мора да има свое име.
- Сите предикати мора да бидат еднозначни, т.е. хомонимите треба да се претстават како различни предикати.
- Внатрешното претставување мора да ја изразува внатрешната структура, т.е. редоследот на компонентите мора да биде: субјект (кој го врши дејството во предикатот), објект (кој го трпи дејството на предикатот).

# Експлицитен избор на референците

- Референтна повеќезначност (referential ambiguity)
- Пр: Јован ја фати топката  
(Jovan\_2 fati topka\_17)

# Еднозначни референци

- Зборовна повеќезначност (word-sense ambiguity)
- Јован фати топка.  
(Jovan\_2 fati\_predmet topka\_17)
- Јован фати грип.  
(Jovan\_2 fati\_bolest grip)

# Внатрешна структура

- Функционална структура (functional structure): (предикат, субјект, објект)
- Јован фати топка  
(fati\_predmet Jovan\_2 topka\_17)
- Јован фати зелена топка  
(fati\_predmet Jovan\_2 topka\_17)  
(instanca topka\_17 topka)  
(boja topka\_17 zelena)

## 2. Процедурално претставување (procedural reasoning)

- Користи симулација за да одговори на прашањата (експертни системи) или да ги решава проблемите (активноста на робот)
- Производни системи од правила од обликот  
IF...THEN... правила



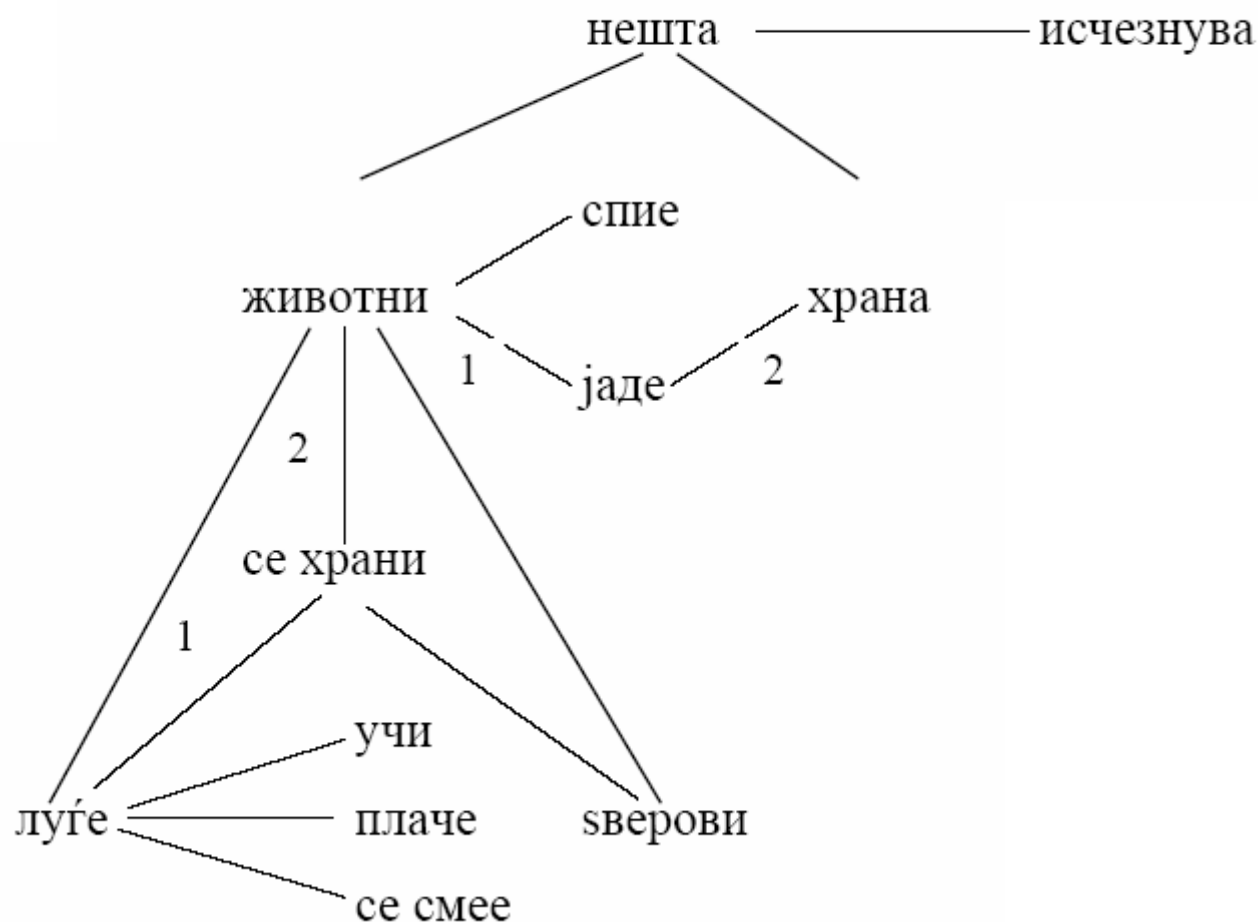
# Својства на производните системи

- Се базираат врз паровите услов-акција, кои се викаат производни правила (production rules) или продукции (productions).
- Основни компоненти:
  - База правила (rule base) која ги содржи производните правила
  - Баферска податочна структура наречена контекст (context)
  - Интерпретатор (interpreter) за контрола на активноста на системот

### 3. Мрежни репрезентациски шеми

- Оваа класа на претставувања користи претстава во форма на граф во кој јазлите претставуваат објекти или концепти во доменот, а гранките претставуваат релации или асоцијации
  - семантички мрежи, концептуални зависимости, концептуални графови, онтологии

# Пример на хиерархиска таксономија

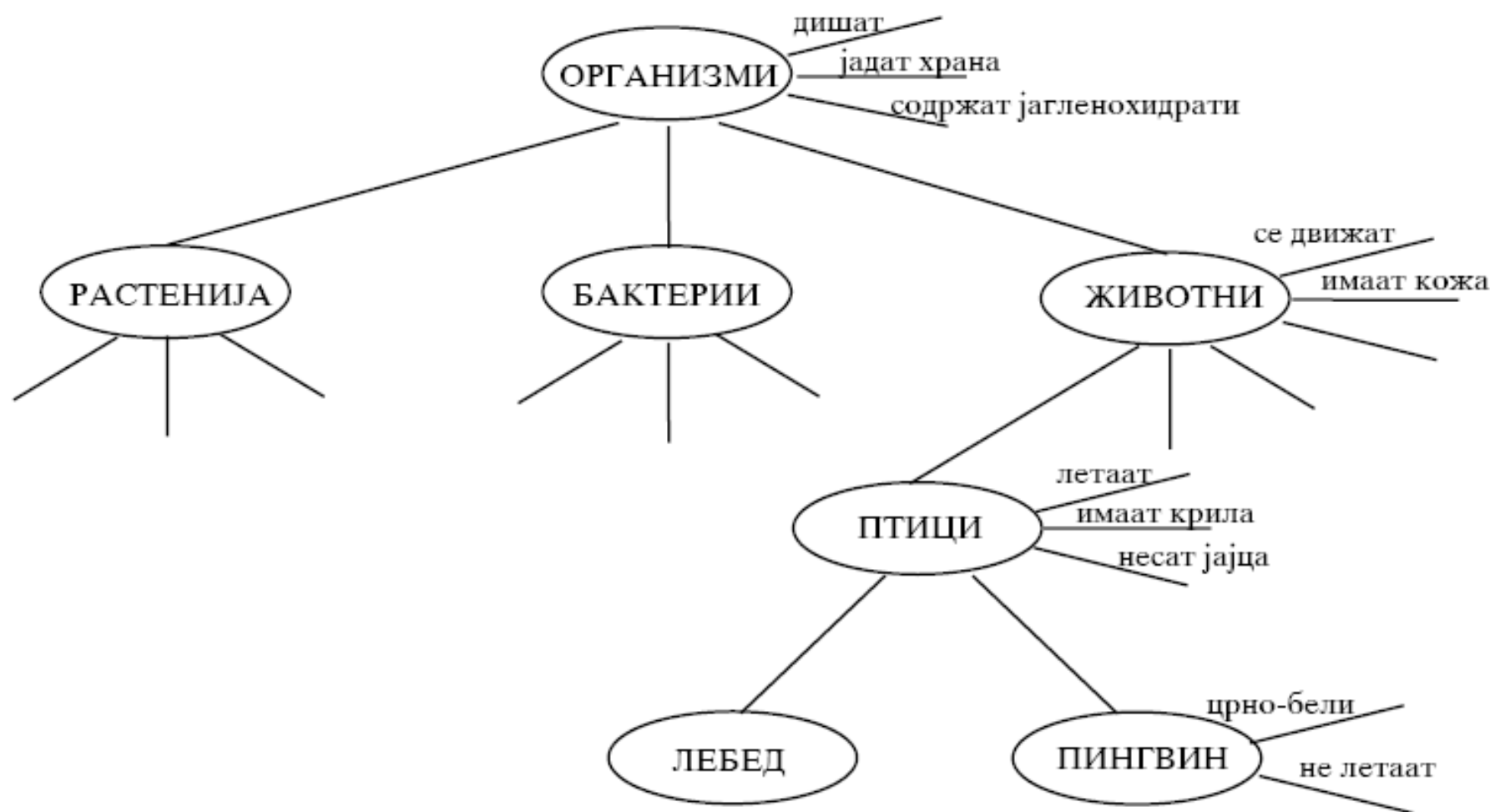




# Семантички мрежи

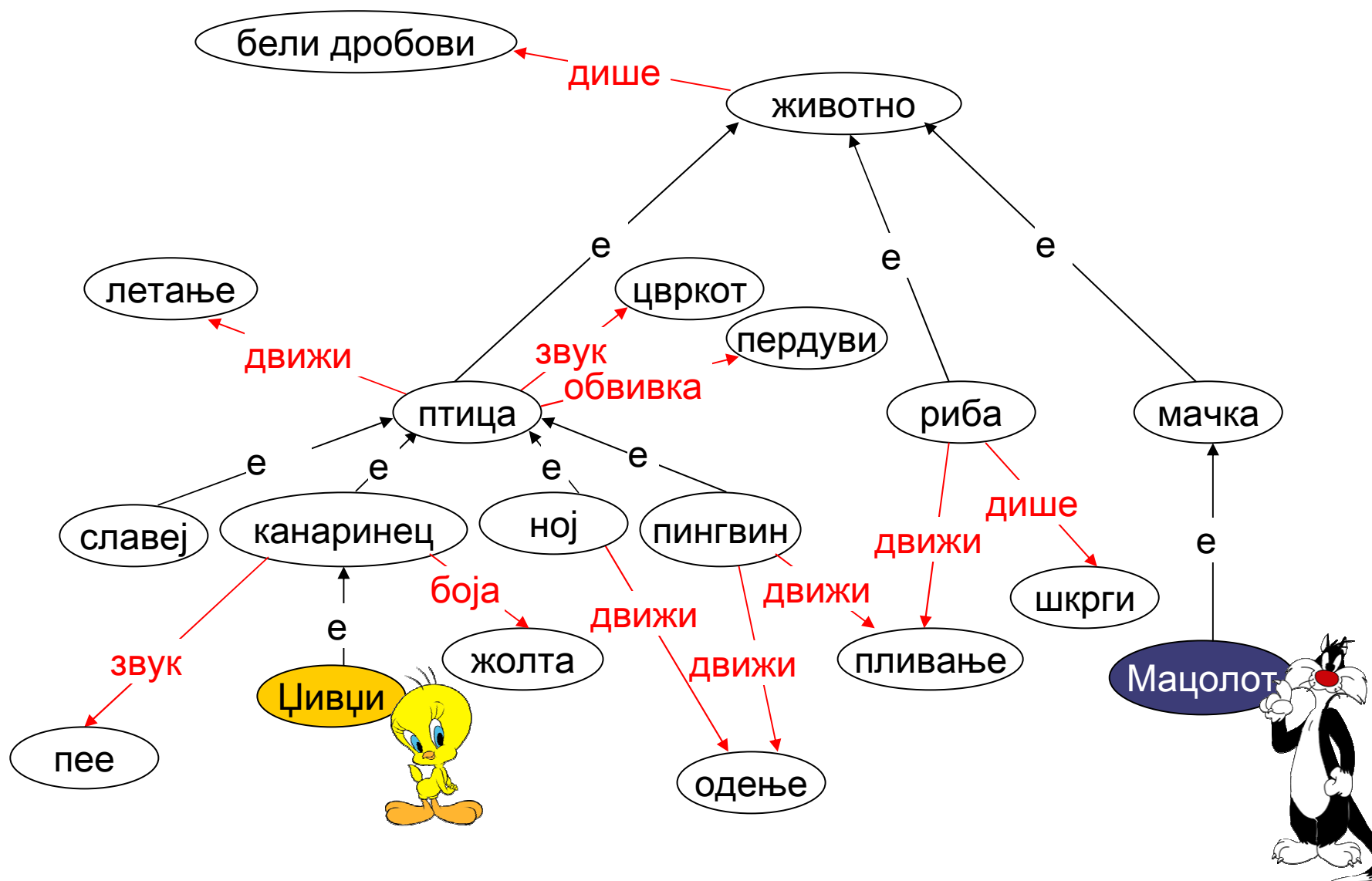
- Мрежа во која:
  - јазлите ги претставуваат објектите, концептите и настаните
  - врските меѓу јазлите ги претставуваат заемните односи
- И јазлите и лаците се именувани.
- Семантичките мрежи експлицитно ги откриваат односите.

# Пример на семантичка мрежа со хиерархиска таксономија

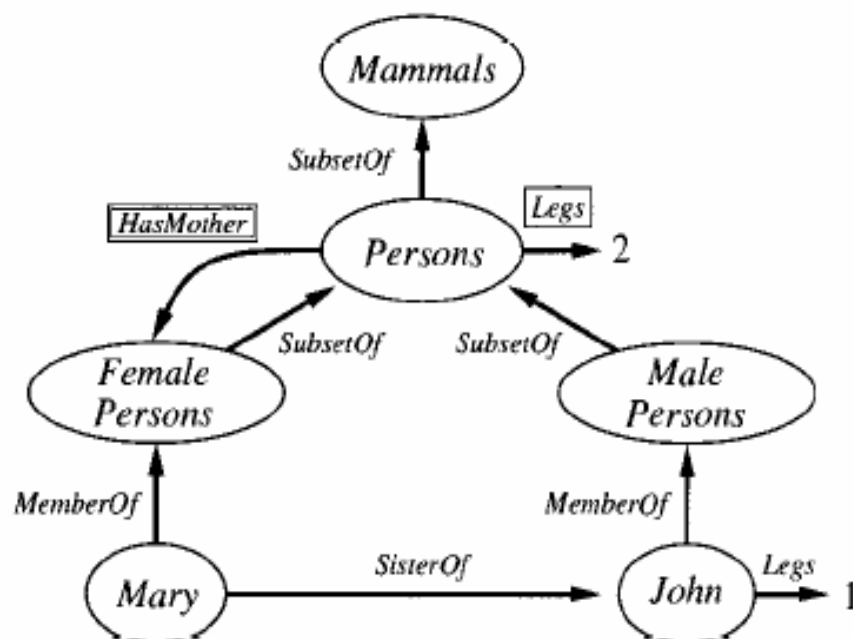


Слика 1. Концепуална хиерархија преставена со СМ

# Друг пример на семантичка мрежа

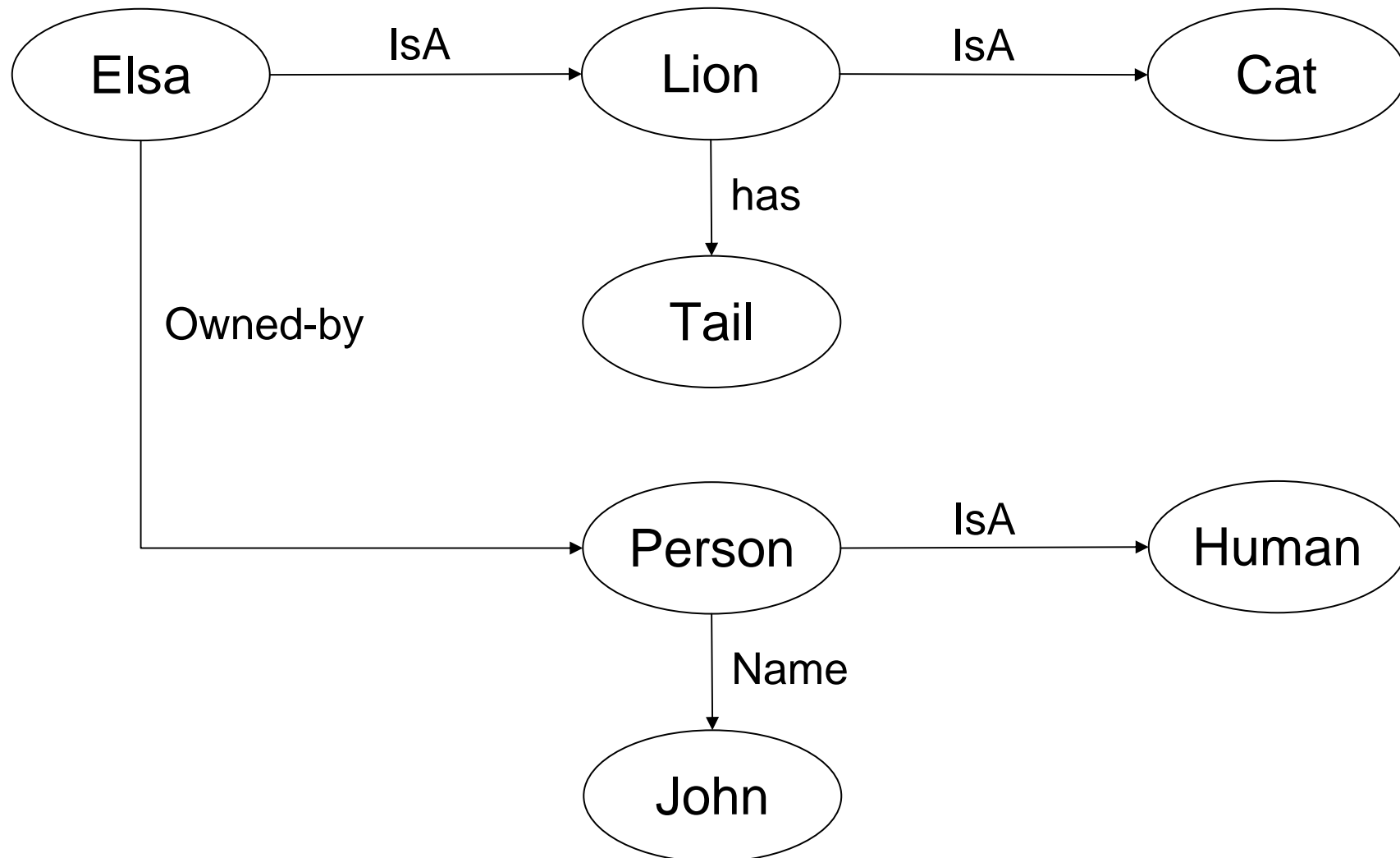


# Пример на семантичка мрежа со четири објекти и четири категории



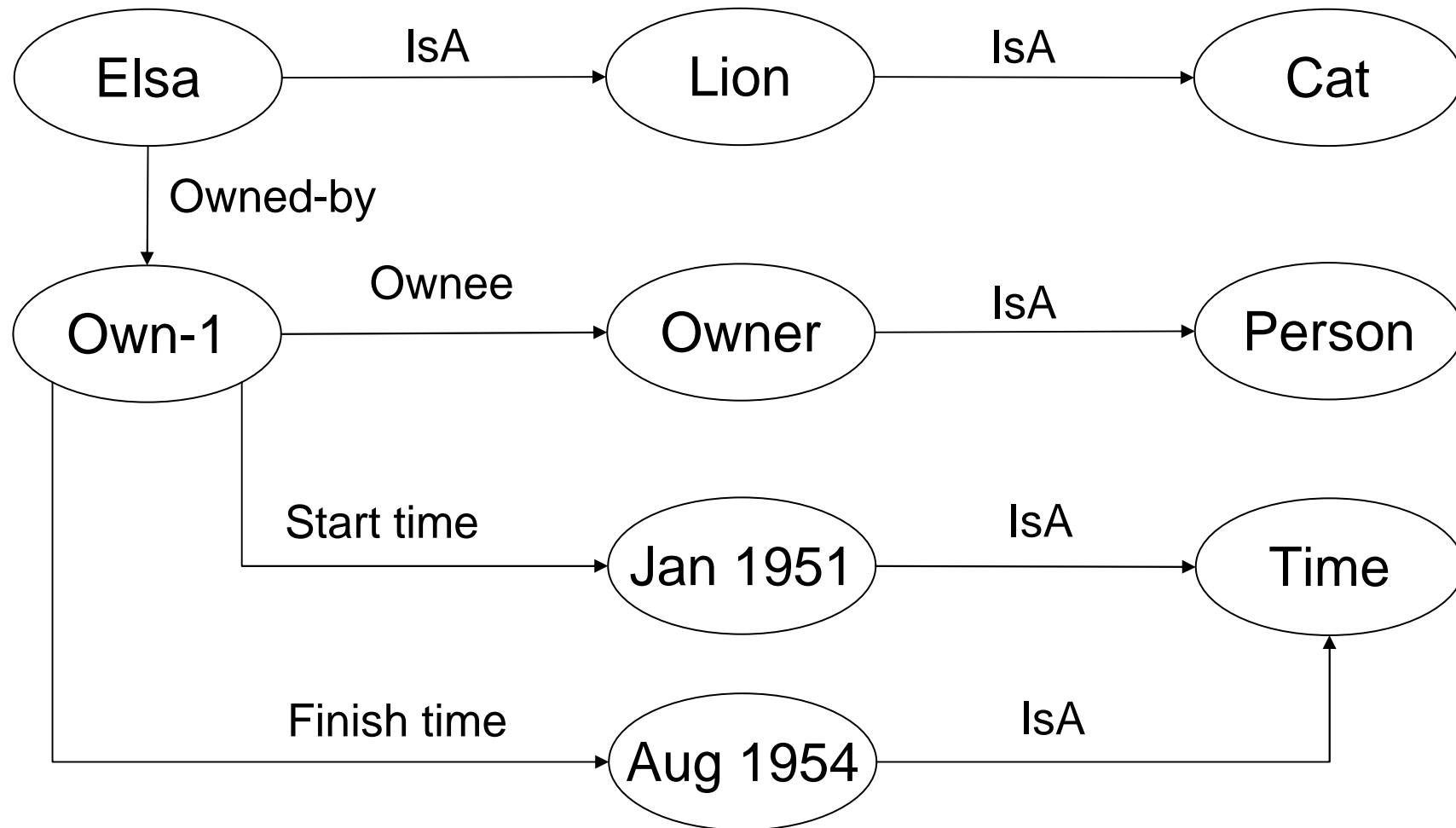
**Figure 10.9** A semantic network with four objects (John, Mary, 1, and 2) and four categories. Relations are denoted by labeled links.

# Семантичка мрежа – пример 5



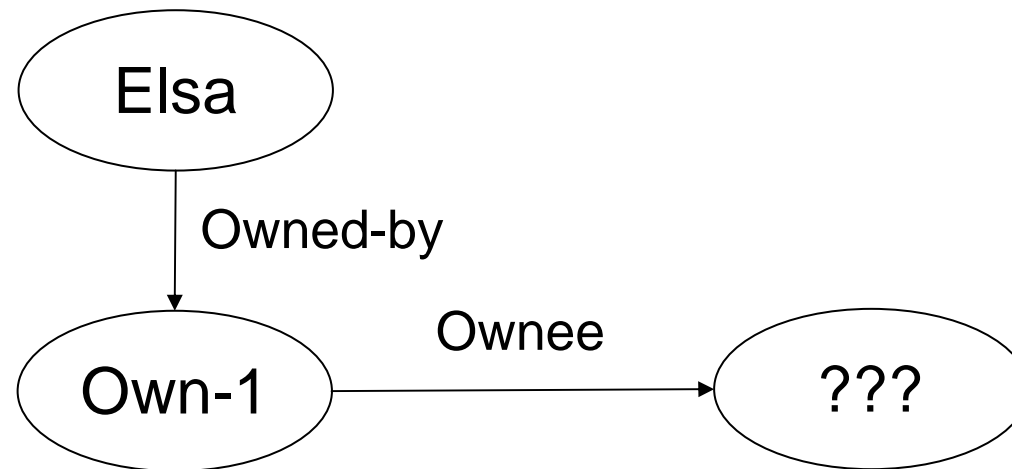


# Семантичка мрежа – пример 6



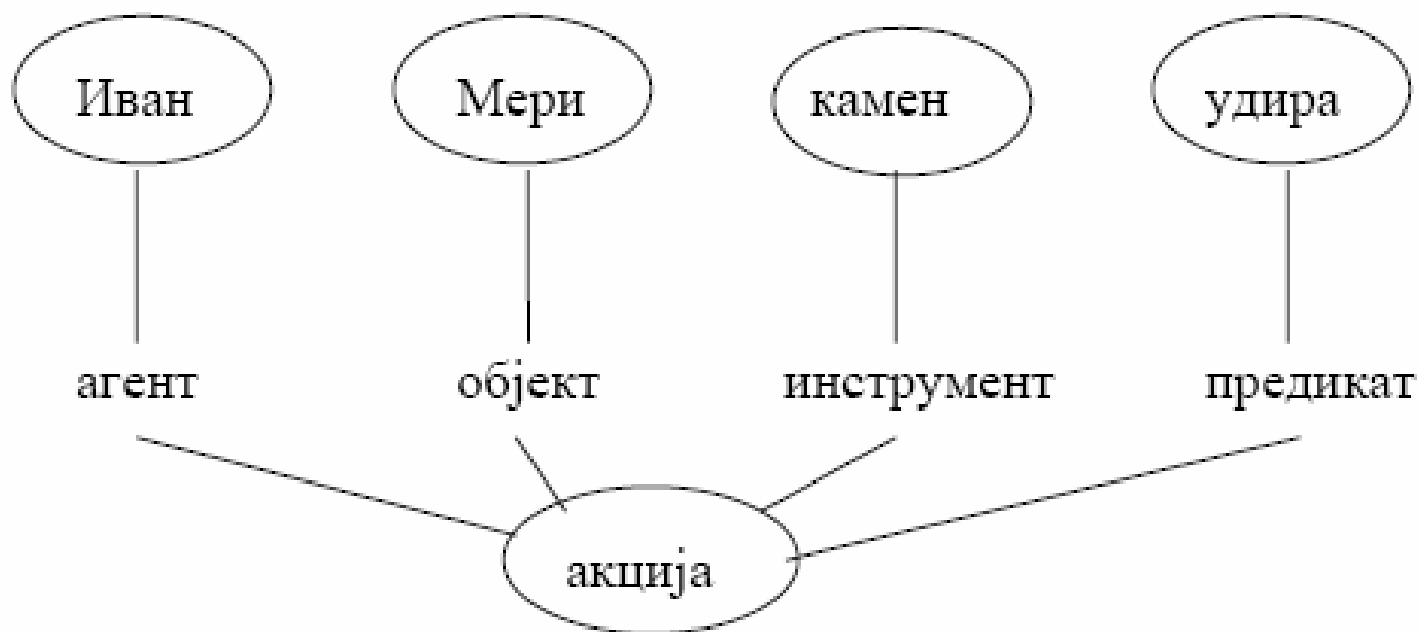
# Изведување кај семантичките мрежи

- Изведување со совпаѓање на шаблони (template matching)
- За да се одговори на „Кој ја поседува Елса?“ треба да се бара подграфот



## Пример 7. Семантичките мрежи и претставувањето на речениците

Иван ја удри Мери со камен.

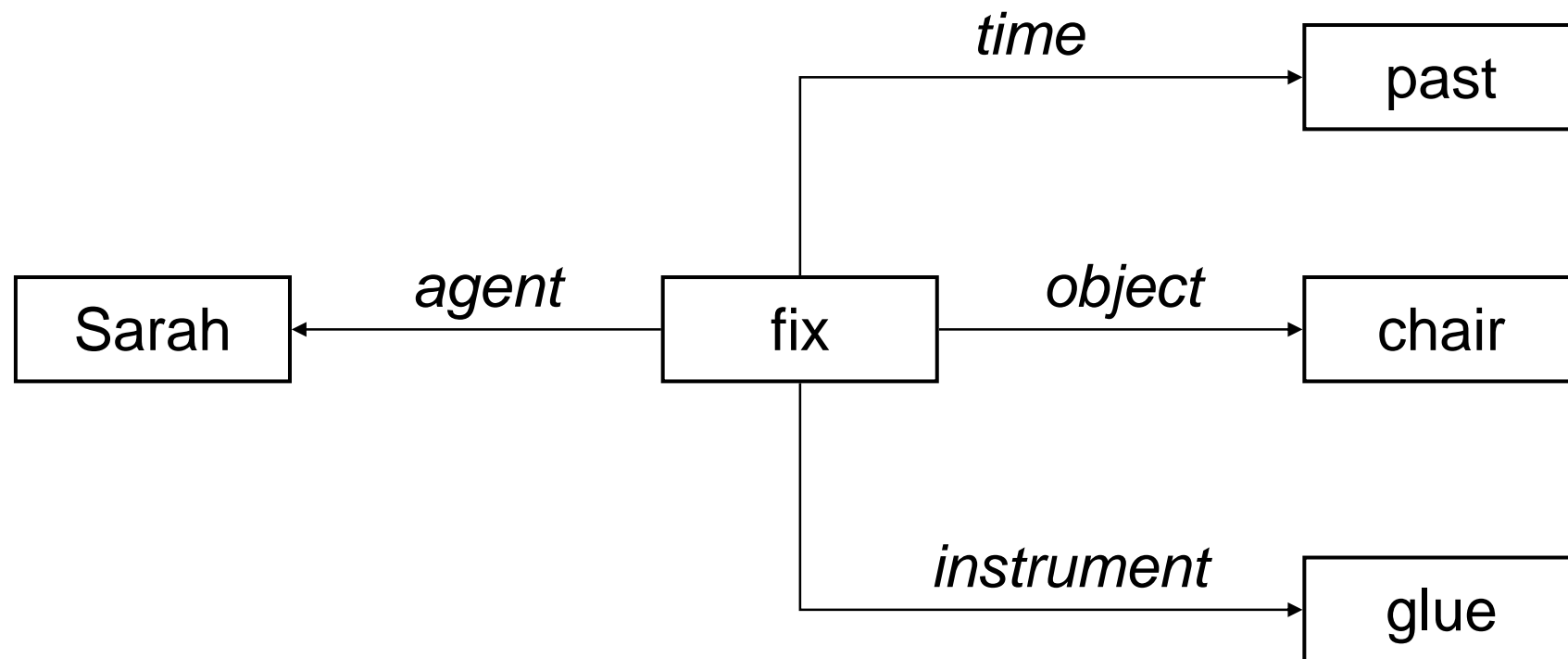


# Стандардизација на релациите во мрежата

- Simmons (1973) – глаголски ориентирана претстава – врските ги одредуваат улогите кои именките и именските фрази ја играат во акцијата на реченицата – case frame структура. Релации: агент, објект, инструмент, локација и време.
- Претстава на знаењето независна од начинот на кој е формулирана реченицата, па дури и јазикот на кој е кажана.

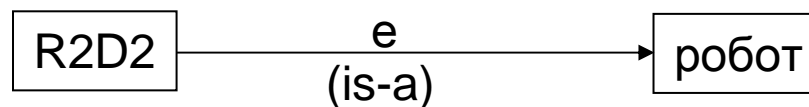


“Sara fixed the chair with glue.”



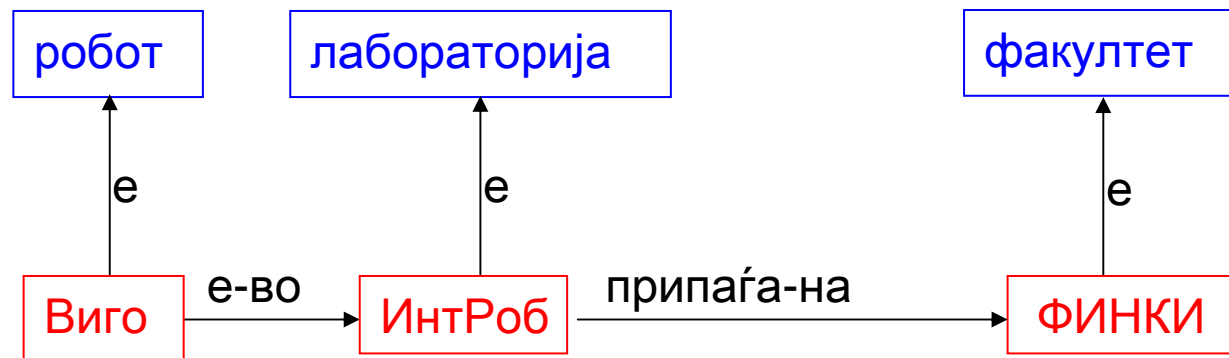
# Семантички молекули

- Семантичка молекула (мрежа, сплет) е граф чии јазли претставуваат објекти, концепти или особини, а чии гранки ги опишуваат односите (релациите) помеѓу нив
- Тоа е мала семантичка мрежа за епизодна меморија (на конкретни настани и ситуации)



# Пример на семантичка молекула

- Роботот Виго се наоѓа во лабораторијата ИнтРоб на факултетот ФИНКИ



# Типови на јазли

- Јазлите во една семантичка молекула можат да бидат поделени на:

- ☐ Генерички (општи)

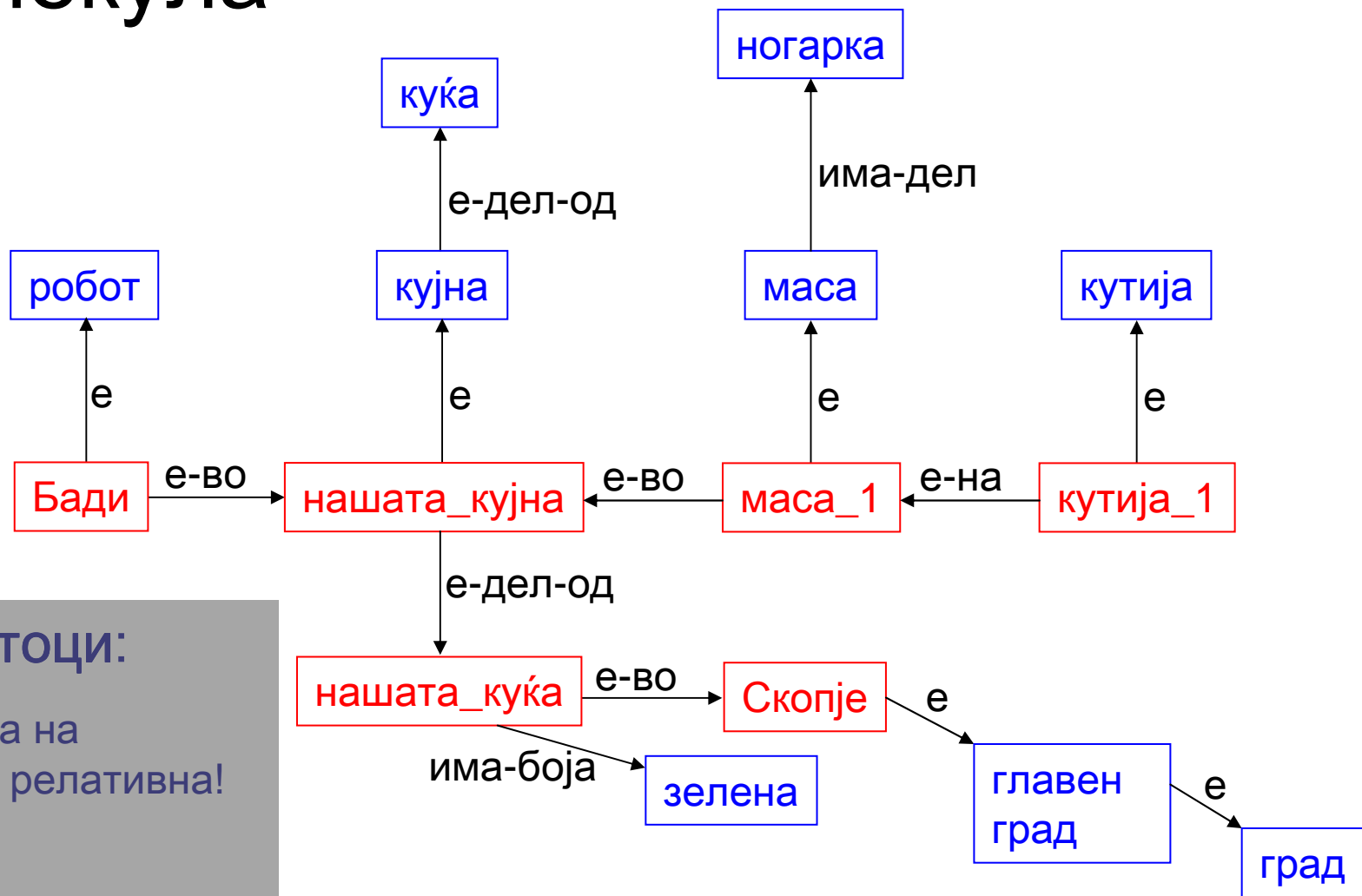
- Обично се во внатрешноста на мрежата

- ☐ Индивидуални

- Обично се на периферијата од мрежата



# Посложен пример на семантичка молекула



## Недостатоци:

Претставата на  
знаењето е релативна!

Не постои  
стандардизирано  
множество релации!

# Стандардизација на односите ...

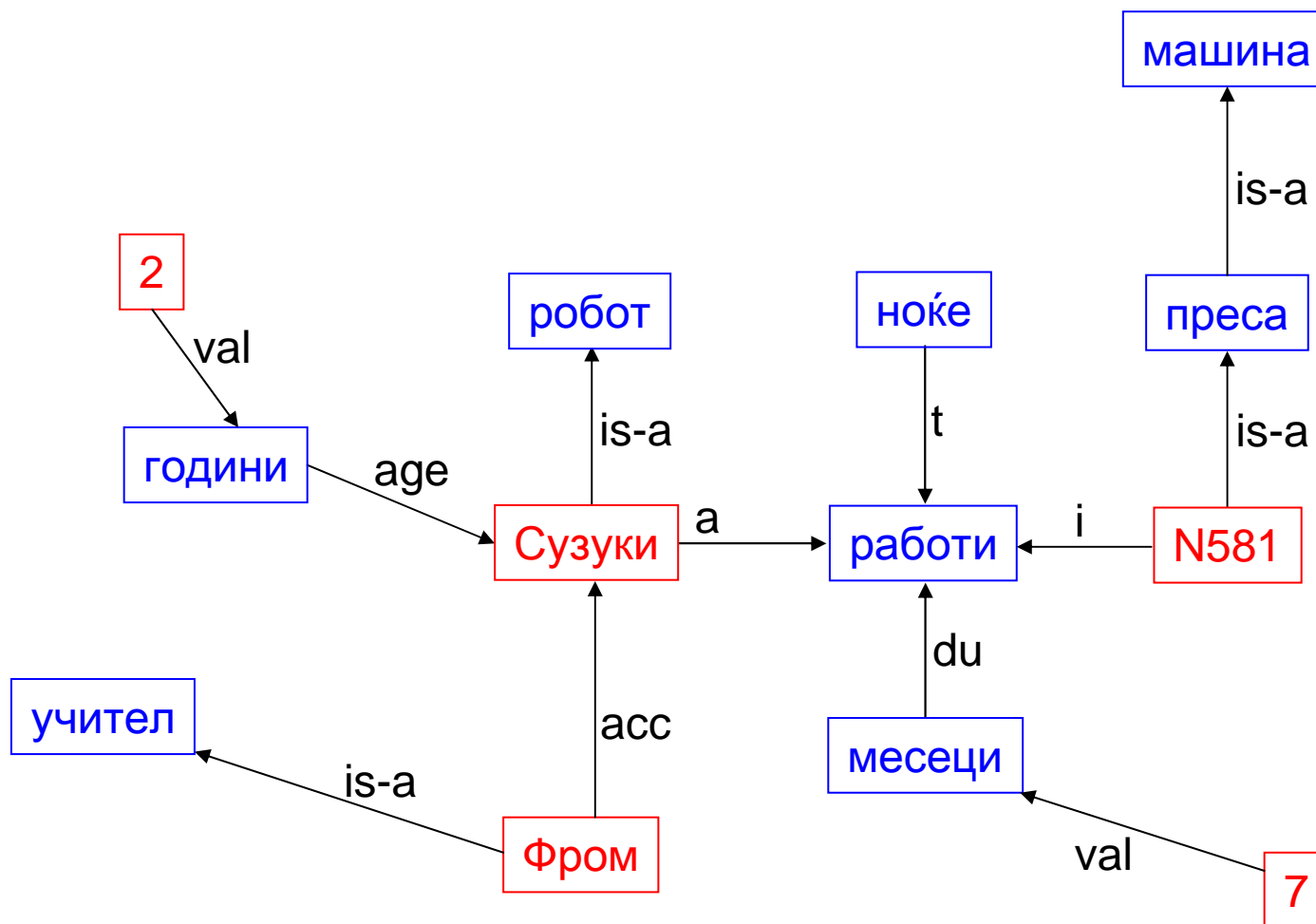
- Предлог преку падежната техника која е јасно изразена кај некои природни јазици

Падеж	Својство на објектот или акцијата	Релевантно прашање	Стандардна релација	Стандардна ознака
номинатив	иницијатор	кој?	agent	a
генитив	припаѓа	чиј?	belong	b
датив	цел	кому?	affected	aff
акузатив	предмет	кого? што?	object	o
аблатив 1	начин	како?	way	w
аблатив 2	време	кога?	time	t
аблатив 3	придружен	со кого?	accompanied	acc
инструментал	инструмент	со што?	instrument	i

# ... Стандардизација на односите

Падеж	Својство на објектот или акцијата	Релевантно прашање	Стандардна релација	Стандардна ознака
локатив	место	каде?	place	p
локатив 1	извор	од кој? од каде?	source	s
локатив 2	правец, одредиште	кон каде?	destination	d
	причина	зошто?	reason	re
	траење	колку долго?	duration	du
	старост	колку стар? од кога?	age	age
	вредност	колку? која мерка?	value	val

# Пример со падежни односи



Кое знаење може да го протолкувате од оваа претстава?

# Дискусија за Е-релацијата

- Може да поврзе:
  - два генерички јазли  
(„е-подмножество-од-множеството“)
  - индивидуален со генерички јазел  
(„е-елемент-од-множеството“)
- Формира природна хиерархија
- Сличност со релацијата „има“
- Наследување на својствата и исклучоците

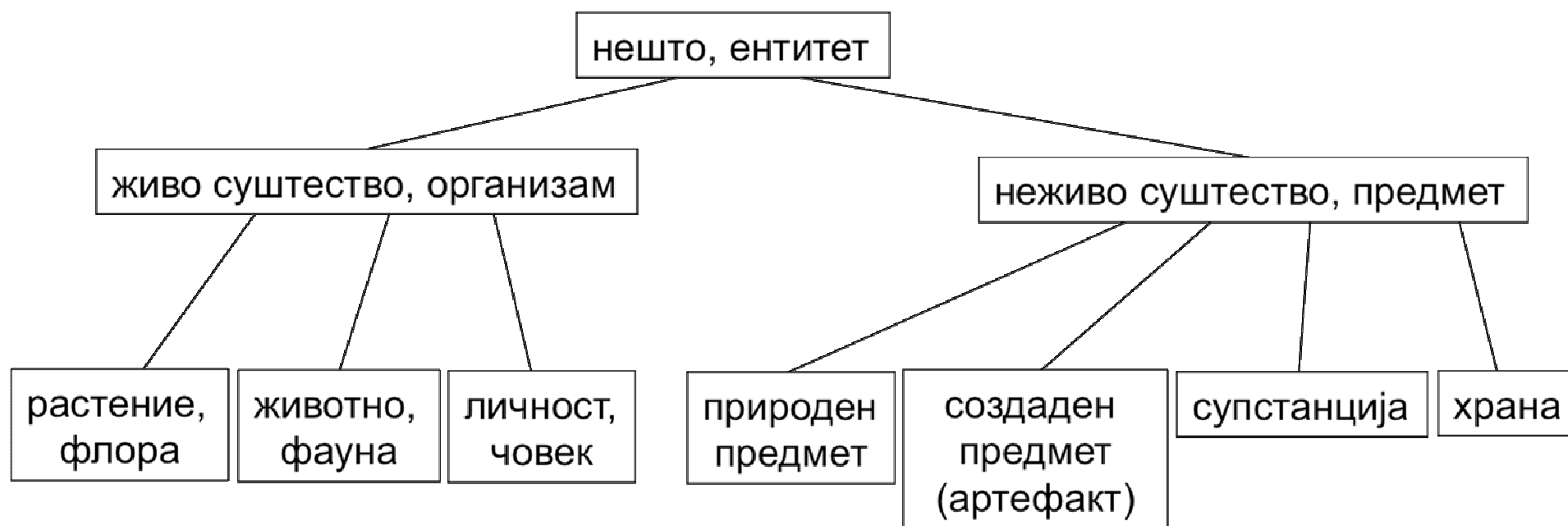
# WORDNET

- Најголемата семантичка мрежа за англиски јазик
- Од Универзитетот Принстон
- Хиерархијата на наследства е најчесто 6-7 нивоа, но и преку 10 за некои специфични категории

# WORDNET

- Три посебни бази на податоци, составени од множество од леми, секое аотирано со множество од различна смисла
  - именки (117.097 WordNet 3.0)
  - глаголи (11.488 WordNet 3.0)
  - придавки и прилози (22.141 WordNet 3.0)
- Во просек именките имаат 1,23 разлики на смислата, а глаголите 2,16 разлики на смислата
- Некои имаат и по 15 до 20 различни значења

# Врвот на хиерархијата во WORDNET (2004 г.)





# Најважни семантички односи (според Википедија и во Ворднет)

- **Синоними (synonym)**
  - А значи исто што и В
- **Антоними (antonym)**
  - А значи спротивно на В
  
- Според синонимија во WordNet  
концептите се организирани во  
множества на синоними, т.н. synsets

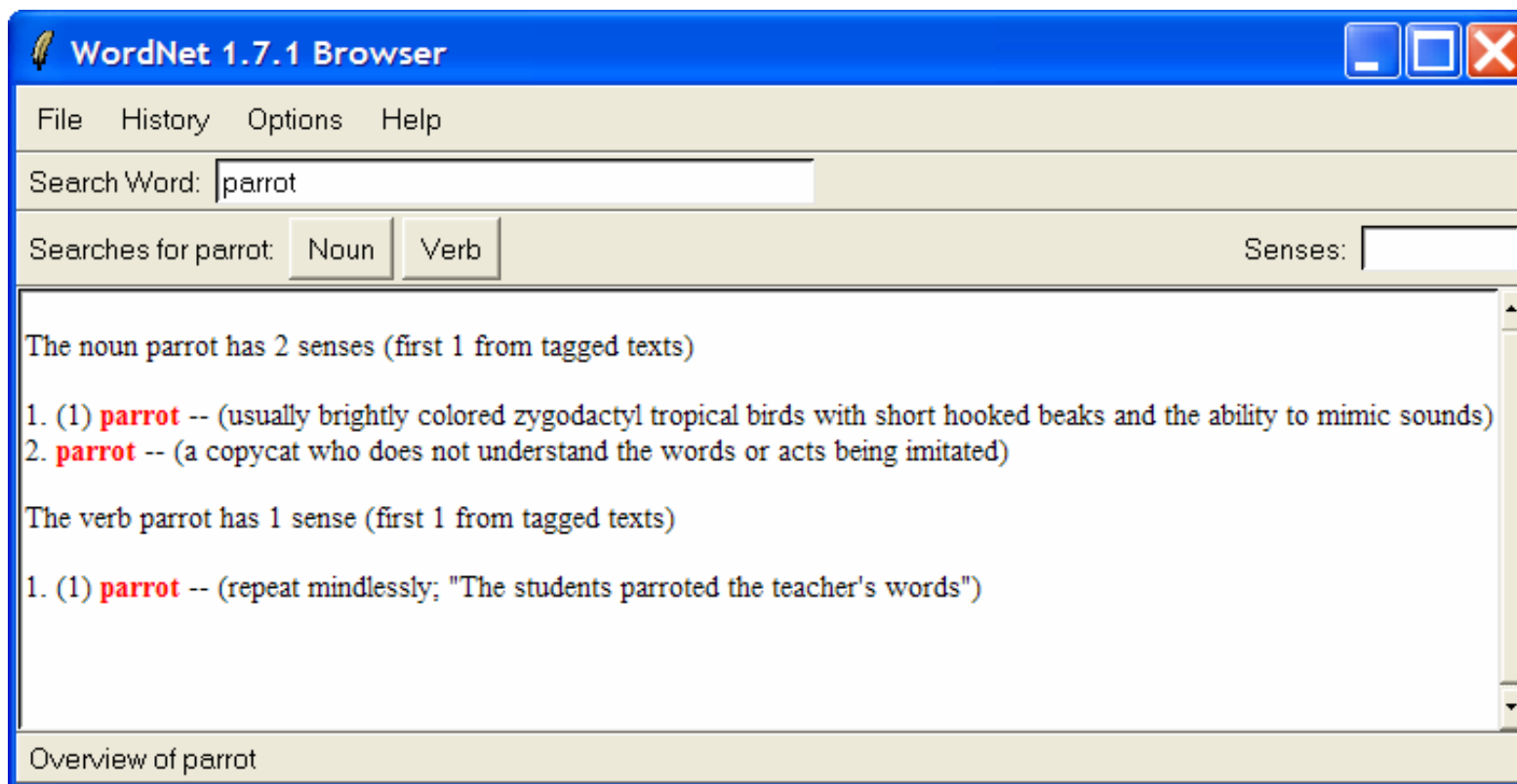
# Најважни семантички односи (според Википедија и во Ворднет)

- **Хиперними (hypernym)**  
(папагалот е вид на птица, птица во однос на папагал е хиперним)
- **Хипоними (hyponym)**  
(ара е вид на папагал, ара во однос на папагал е хипоним)
- **Холоними (holonym)**  
(папагалот има предувии, папагал во однос на пердуви е холоним)
- **Мероними (meronym)**  
(крило е дел од папагал, крило во однос на папагал е мероним)

# Примери со WORDNET Browser

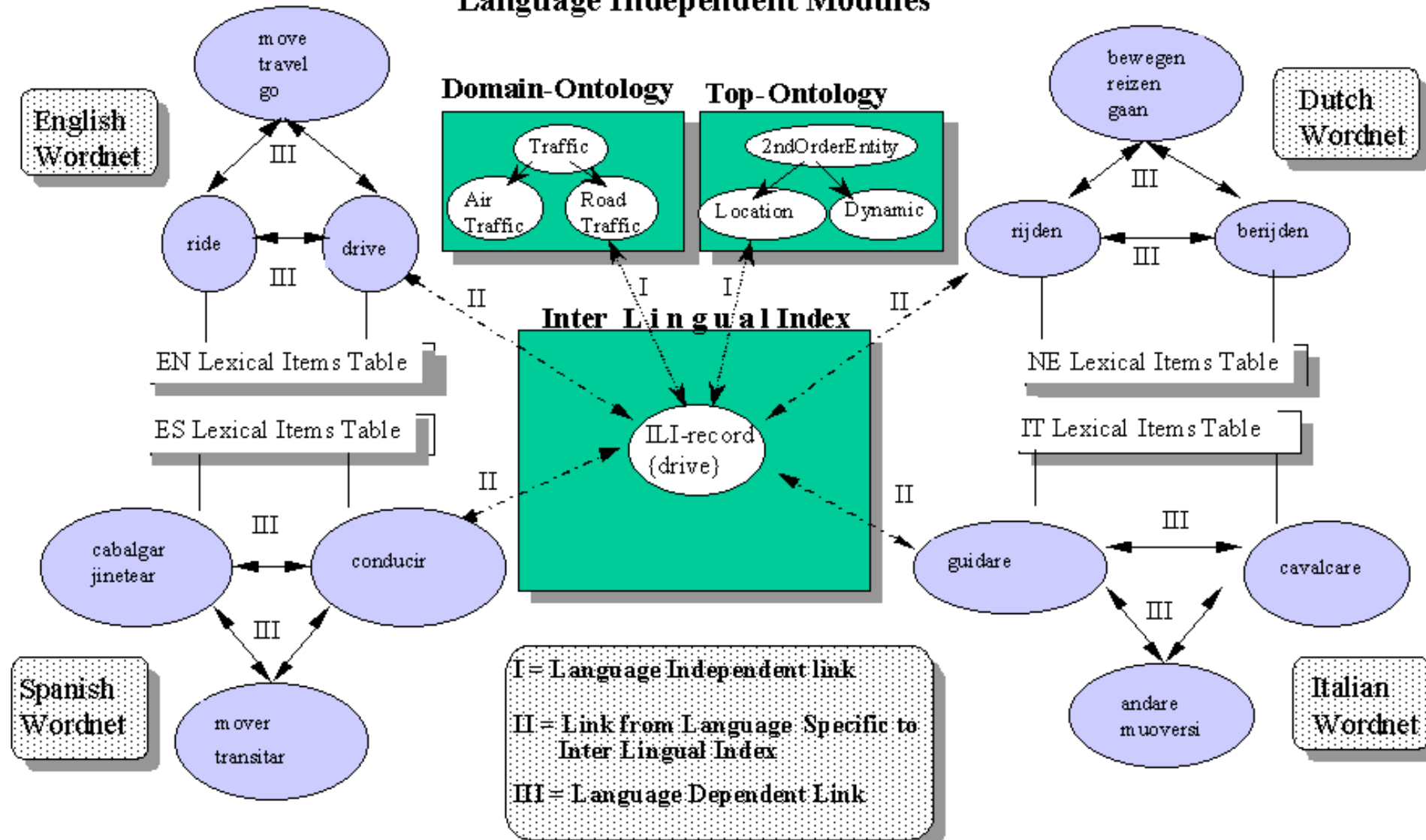
- parrot
- entity (thing)
- get

Онлајн-верзија:  
<http://wordnetweb.princeton.edu/perl/webwn>



# Архитектурата на EURO-WordNet

## Language Independent Modules



Сега веќе постојат и многу други Ворднети, <http://globalwordnet.org/wordnets-in-the-world/> некои се целосно отворени (бесплатни), некои делумно, некои се само комерцијални.

# Концептуална теорија на зависност

- Roger Schank – амбициозен обид да се моделира длабоката семантика на природните јазици – концептуална теорија на зависност (Conceptual Dependency Theory)
- Предлага 4 примитивни концептуализации:
  - ACTs – акции
  - PPs – објекти (picture producers)
  - AAc – модификатори на акции (action aiders)
  - PAs - модификатори на објекти (picture aiders)

# Семантичките примитиви на Шанк (Schank) \*

- (1) ATRANS трансфер на апстрактен однос, како поседување, сопственост или контрола
- (2) PTRANS трансфер на физичка локација на објект
- (3) PROPEL примена на физичка сила над објект
- (4) MOVE движење на дел од телото на животно од самото тоа животно
- (5) GRASP фаќање предмет од страна на учесник
- (6) INGEST прифаќање објект од животно
- (7) EXPEL исфрлување од тело на животно надвор
- (8) MTRANS трансфер на ментална информација меѓу животни или меѓу свесни процесори, долготрајна меморија или сетилни органи на животните
- (9) MBUILD конструкција на нова информација од страна на животно
- (10) SPEAK активност на производство звуци
- (11) ATTEND активност на следење или насочување на сетилото кон стимулот

# Пример за претставување на знаењето во Шанковиот модел

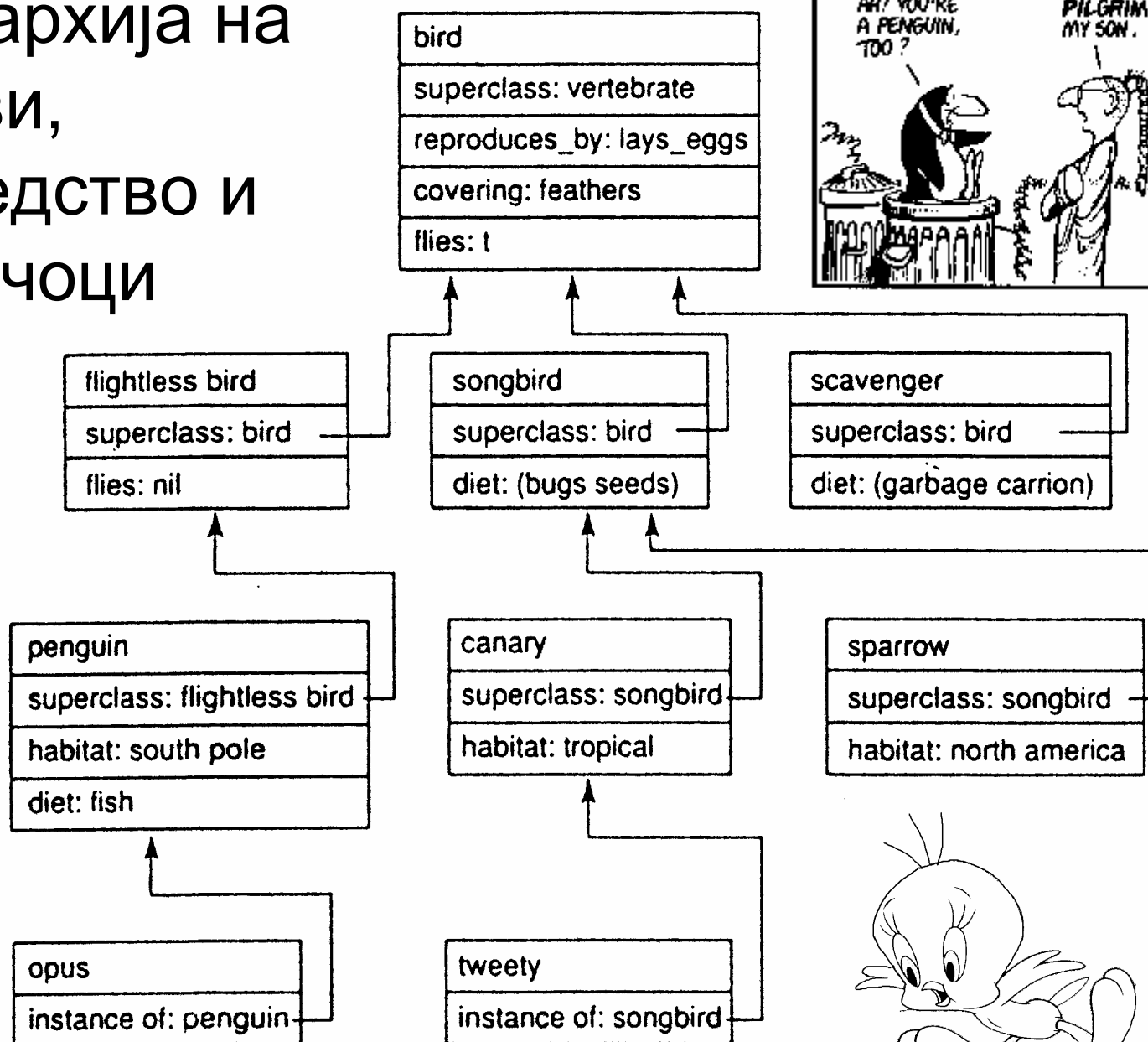
- Јован ја удри Мери.
- Јован ја примени силата на физичкиот објект од Јован кон Мери причинувајќи состојба на физички контакт меѓу објектот и Мери.

# Наследувањето на својствата кај семантичките мрежи

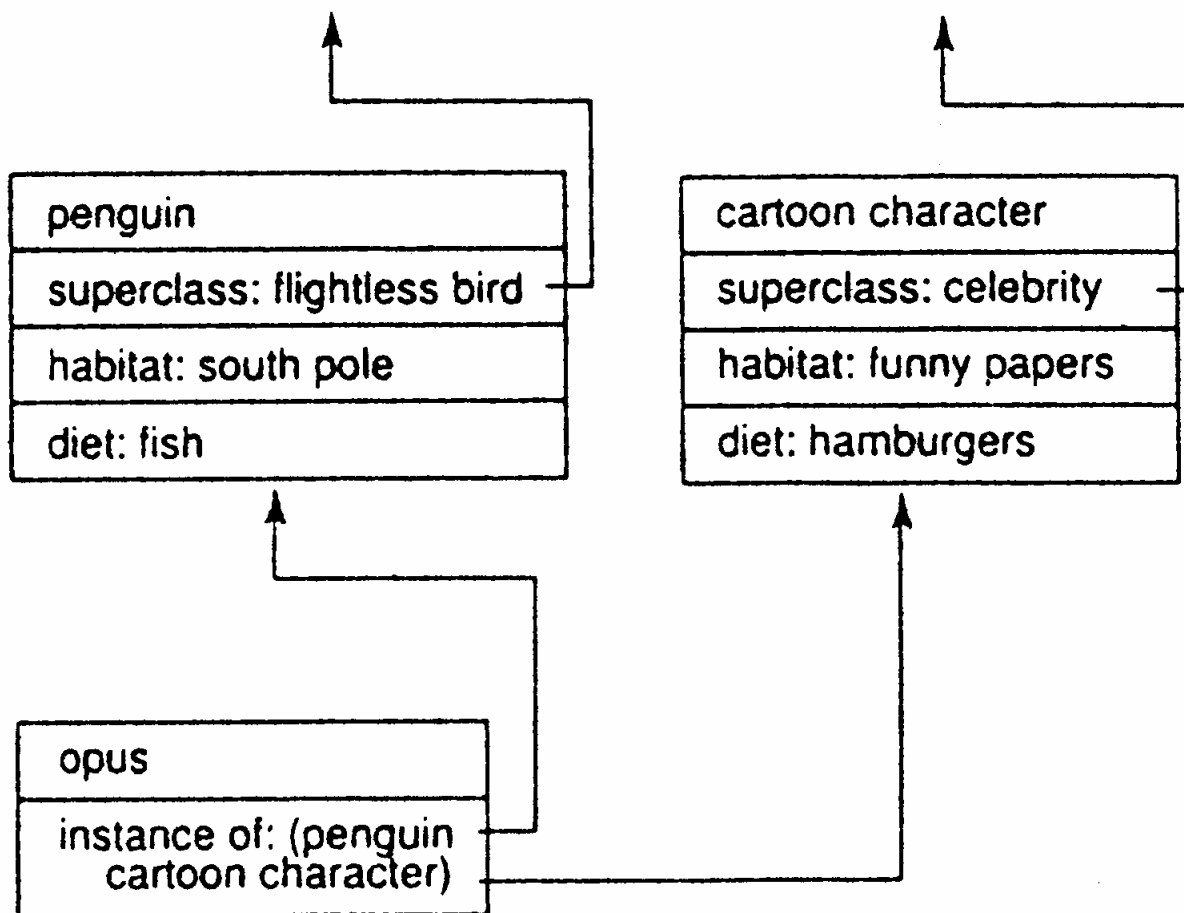
- Наследувањето на својствата може да биде во противречност со различните типови јазли.
- Решение: откажување на наследувањето (cancelation of inheritance).
- Пр: Нека класата птици го има својството дека тие летаат. Појавата на пингвинот ја разложува класата птици на две поткласи, од кои само едната го наследува својството, а другата добива спротивно својство.



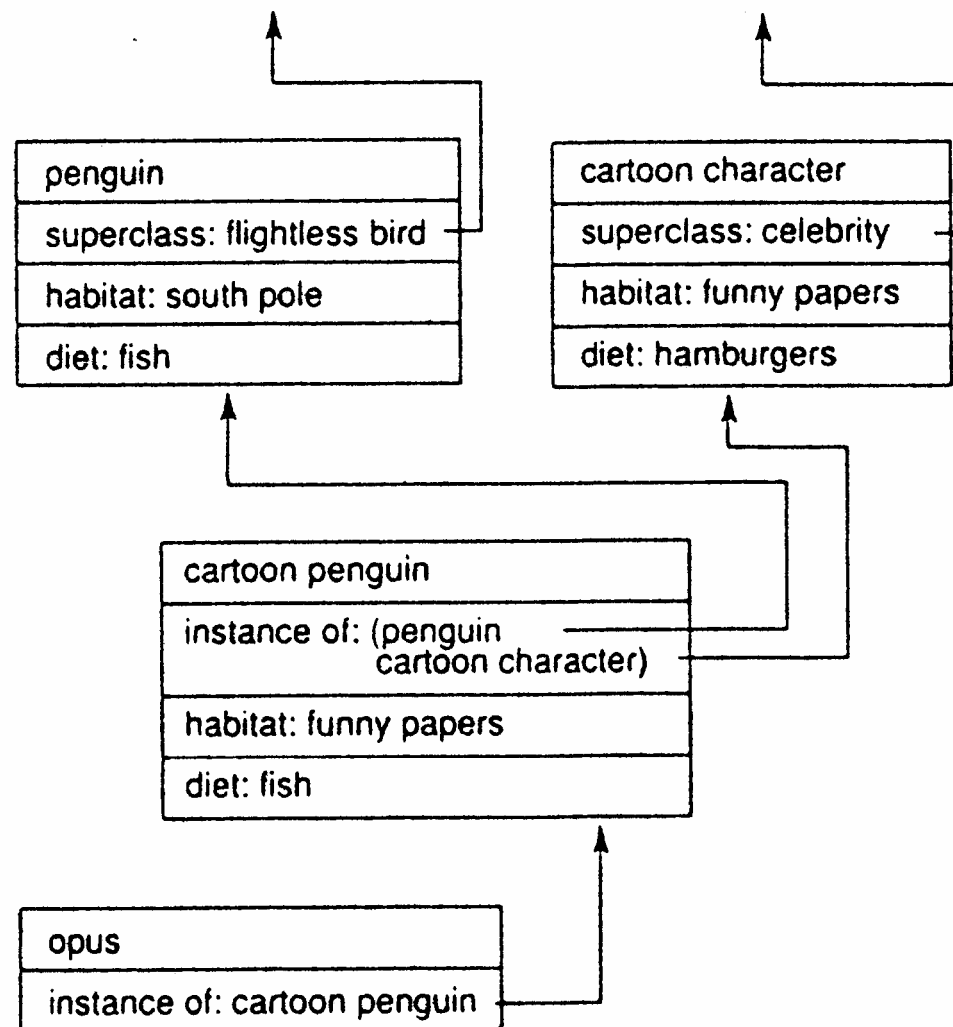
# Хиерархија на типови, наследство и исклучоци



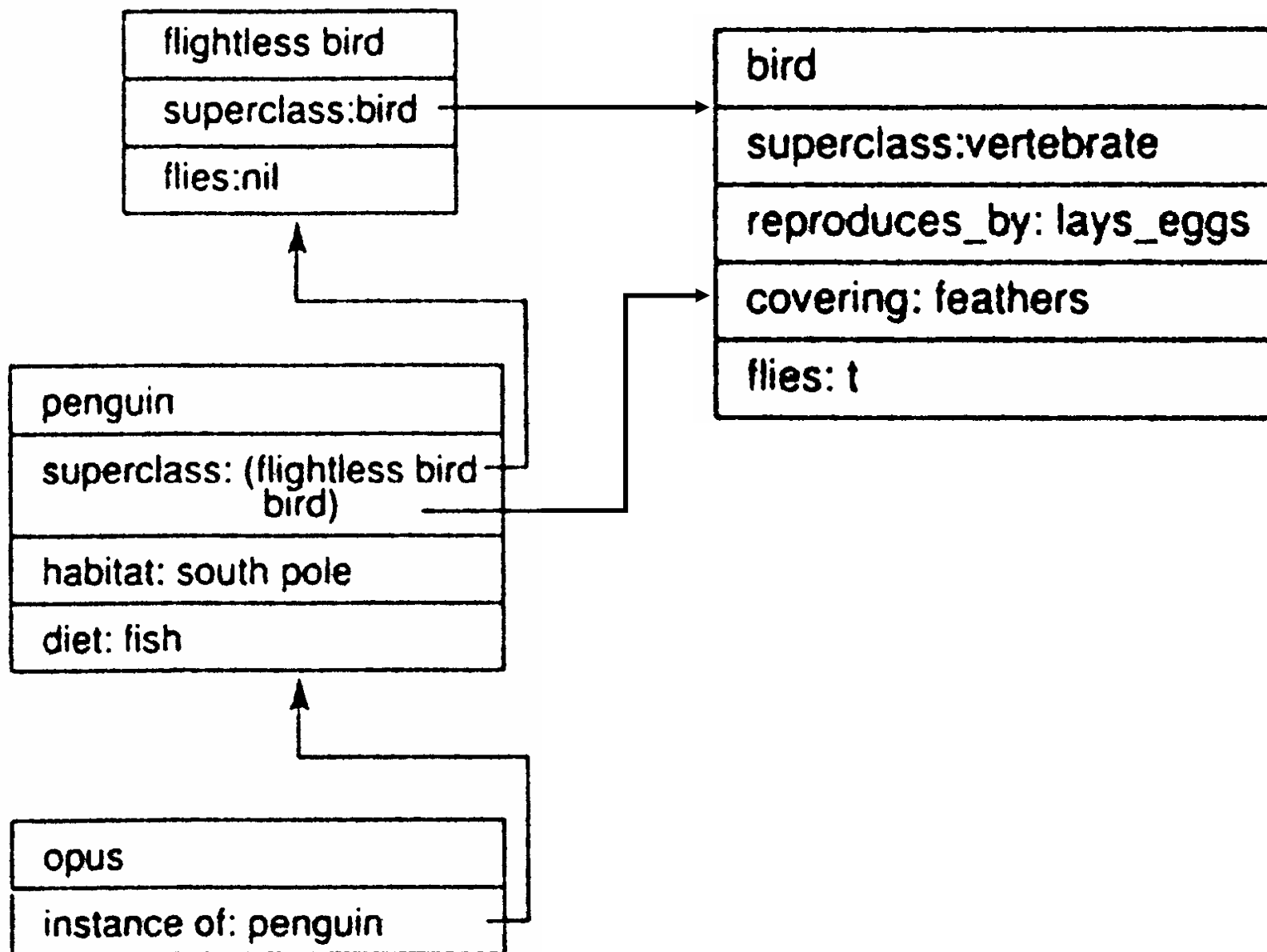
# Повеќекратно наследување



**Figure 9.27** An ambiguous multiple inheritance situation.



**Figure 9.28** Introduction of a new class to resolve ambiguity.



# Онтологији

- Онтологијата е наука која се занимава со суштината на концептите кои постојат во реалноста.
- Под онтологија се подразбира претставувањето на најопштите концепти кои се карактеристични за повеќе различни домени.
- WordNet и Сус се типични примери за таканаречените робусни онтологији.

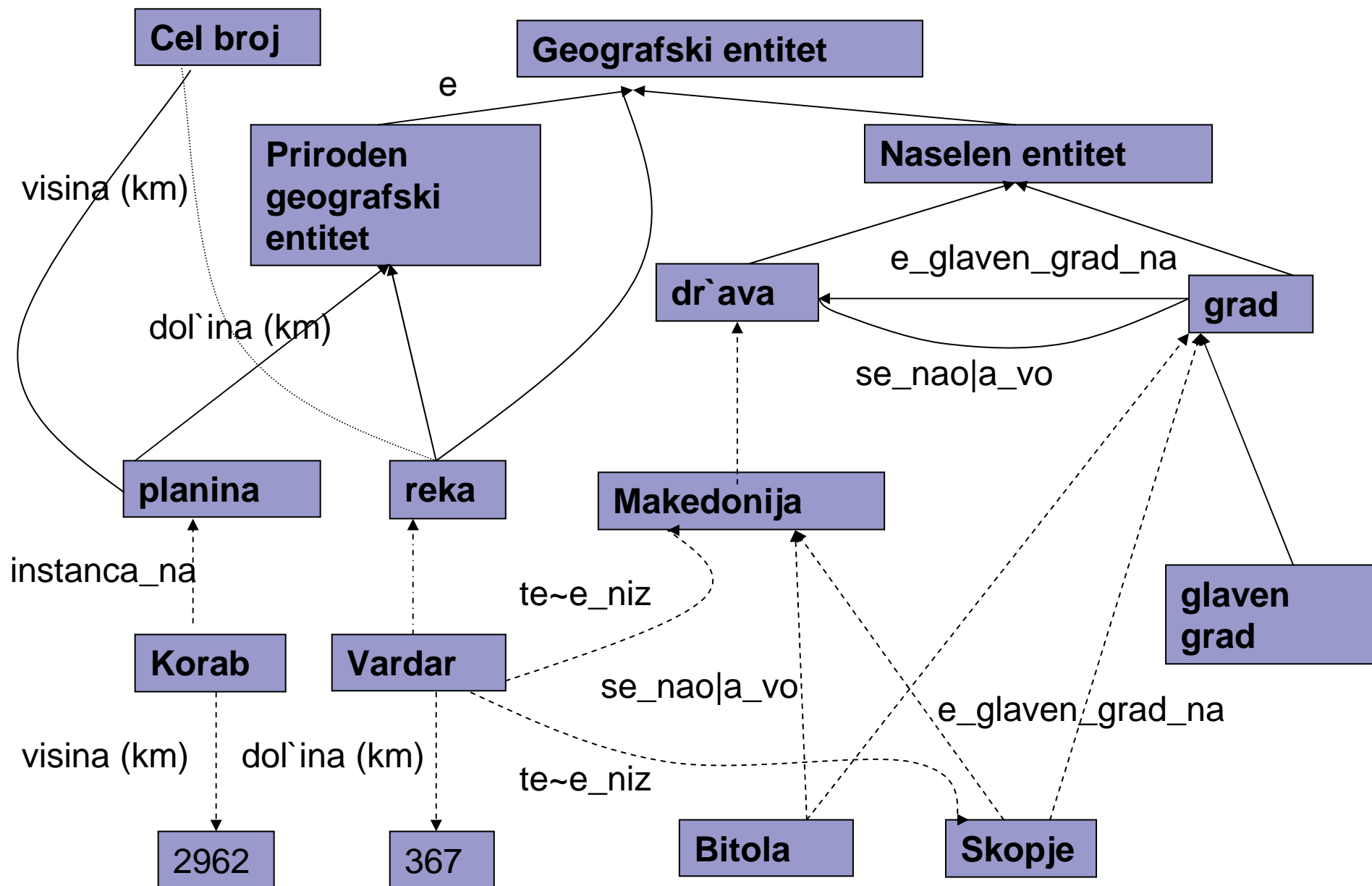
# Онтологијата во информатиката

- Онтологијата е начин за прикажување на множества концепти и односите кои се воспоставени меѓу тие концепти.
- Онтологијата не е ограничена на мал број домени.
- Практичната примена на онтолошките концепти се вика апликативна онтологија.
- Постојат голем број програмски јазици кои ги поддржуваат онтологиите.

# Основните компоненти на онтологијата

- Единки: објекти или инстанци
- Класи
- Атрибути
- Релации
- Функциски терми
- Правила
- Аксиоми
- Настани

# Primer na ontologija





## 4. Структурни репрезентациски шеми

- Претставува проширување на мрежната репрезентација со дозволување секој јазел да биде комплексна податочна структура (рамка) составена од именувани атрибути со придружени вредности. Овие вредности можат да бидат примитивни нумерички или симболички податоци, покажувачи кон други рамки па дури и процедури за изведување одредена задача – скрипти, рамки, објекти

# Рамки (frames)

- Оригинално воведени од Minsky во 1975
- Рамките се способни да поседуваат метазнаење
- Тие го организираат знаењето во форма на целини кои ги објаснуваат ситуациите (обично стереотипните) или објектите.
- Родителска рамка: мемориска структура која ги содржи сите слотови (slots) кои опишуваат еден објект или ситуација.
- Слотовите во суштина се својствата на објектот или ситуацијата.

# Пример од Barr & Fiegenbaum, 1981

## CHAIR Frame

Specialization_of:	FURNITURE
Number_of_legs:	an integer (default = 4)
Style_of_back:	straight, cushioned, . . .
Number_of_arms:	0, 1 or 2

## John's CHAIR Frame:

Specialization_of:	CHAIR
Number_of_legs:	4
Style_of_back:	cushioned
Number_of_arms:	0

# Расудувањето кај рамките и сценаријата (scripts)

- Рамките обезбедуваат структура во која новите податоци се интерпретираат преку претходно научени концепти.
- Организацијата на знаењето го олеснува процесирањето водено од очекувањата (expectation driven processing) барајќи ги нештата кои се очекуваат зависно од контекстот во кој се смета дека припаѓаат.

# Кои се функциите кои го поддржуваат расудувањето

- Пополнување слотови (Slot Filler)
- Наследување (If-Needed)
- Програмчиња, т.н. “демони” за процедурално прикачување (Procedural Attachment Demons)

# Рамка со сценарио, пример

Generic RESTAURANT Frame

Specialization-of: Business-Establishment

Types:

range: (Cafeteria, Seat-Restaurant, Wait-To-Be-Seated)  
default: Wait-To-Be-Seated  
if-needed: IF plastic-orange-counter THEN Fast-Food,  
IF stack-of-trays THEN Cafeteria  
IF wait-for-waitress-sign or reservation-made  
THEN Wait-To-Be-Seated  
OTHERWISE Seat-Yourself

Location:

range: an ADDRESS  
if-needed: (Look at the MENU)

Name:

if-needed: (Look at the MENU)

# Продолжение на примерот

## Food-Style:

range: (Burgers, Chinese, American, Seafood, French)

default: American

if-needed: (Update Alternatives of Restaurant)

## Times-of-Operation:

range: a Time-of-Day

default: open evenings except Mondays

## Payment-Form:

range: (Cash, CreditCard, Check, Washing-Dishes-Script)

## Event-Sequence:

default: Eat-At-Restaurant Script

## Alternatives:

range: all restaurants with same FoodStyle

if-needed: (Find All Restaurants with the same FoodStyle)

# Сценариото во рамката (1)

## EAT-AT-RESTAURANT Script

Props:

(Restaurant, Money, Food, Menu, Tables, Chairs)

Roles: (Hungry-Persons, Wait-Persons, Chef-Persons)

Point-of-View:

Hungry- Persons

Time-of-Occurrence:

(Times-of-Operation of Restaurant)

Place-of-Occurrence:

(Location of Restaurant)



# Сценариото во рамката (2)

## Event-Sequence

```
first:      Enter-Restaurant-Script
then:      if (Wait-To-Be-Seated-Sign or
Reservations)
            then Get-Maitre-d's-Attention Script
then:      Please-Be-Seated- Script
then:      Order-Food-Script
then:      Eat-Food-Script unless (Long-Wait) when
            Exit-Restaurant-Angry Script
then:      if (Food-Quality was better than Palatable)
            then  Compliments-To-The-Chef Script
then:      Pay-For-It Script
finally:   Leave-Restaurant Script
```

# Резиме за семантичките мрежи и рамките

- Мрежни формализми
- Поддржани од програмски алатки: FrameShop (<http://www.bagism.com/frameshop/>)
- Тешко ги опфаќаат логичките оператори
- Несоодветни за приказ на нетаксономното знаење
- Компромисно решение: терминолошка логика (terminological logics)
  - хиерархиска структура за приказ на ентитетите, класите и својствата
  - логички приказ на останатите информации

# Главен недостаток на сите овие претставување на знаење

- Не е предложен начин како тие да се учат. Како машините самостојно да ги учат ваквите претстави.
- Нив мора стално некој инженер да ги додава во базата на знаење.
- А човекот, на пример, е способен постојано да измислува, да комбинира или да учи нови категории (концепти)

Оваа слика ја „буди“ (или ја повикува) вашата претстава за „јајце“ (ве потсетува на „јајце“)  
(This image invokes your representation of “egg”)



Оваа слика ве потсетува на „тросед“  
(This image invokes your representation of “sofa”)



# Оваа слика ве потсетува на ???



# Други примери на ад-хок категории (концепти)

- (кошница, чергичка, сендвичи, ноже, чаши, пиво, сок, ...)
  - Работи што се понесуваат со себе на излет
- (Рено-4, казан за печење ракија, систем со звучници, ...)
  - Работи што се нудат за продажба во огласник

# Креативните луѓе секогаш измислуваат или откриваат нови работи

$$\pi = 3,141592\dots$$





# Користена литература

- Artificial Intelligence, A Modern Approach  
2nd edition, Russel and Norvig
- Artificial Intelligence, A New Synthesis, Nils J. Nilsson
- Божиновски С., Вештачката интелигенција,  
Гоцмар, Скопје, 1994



# Прашања?

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Веројатносно расудување

# Од каде потекнува несигурноста?

- Логичките системи со кои се претставуваат проблемите од реалниот живот не се комплетни.
- Но, агентите мора да дејствуваат и во вакви услови.
- Расудувањето во услови на некомплетност, неконзистентност и немонотоност се вика расудување во услови на несигурност.

# Причини за несигурноста според Charles R. Dyer

- Несигурност која се должи на случајноста (на пр. фрлање паричка или коцка): вистинска несигурност
- Отсуство на знаење кое се должи на некомплетноста на системот (на пр. медицинска дијагноза): теоретско незнаење
- Несигурност која се должи на преобемноста на релевантни фактори, или на неможноста да се опфатат сите меѓурекултати: мрзливост
- Несигурност која се должи на недоволно прибрани информации во врска со доменот: практично незнаење

# Трите принципа на несигурноста

1. Принцип на максимизација на несигурноста (Principle of Uncertainty Maximization)
2. Принцип на минимизација на несигурноста (Principle of Uncertainty Minimization)
3. Принцип на непроменливост на несигурноста (Principle of Uncertainty Invariance)

# Максимизација на несигурноста (1)

- Основна идеја: при индуктивното расудување да се искористат сите расположливи информации, но не повеќе од тоа, така што ќе се обезбеди да не се појави ниту една дополнителна информација што е несакајќи додадена.
- Применлив е кај системите во кои е вклучено амплијативното расудување.
- Амплијативно е тоа расудување кај кое претпоставките не ги повлекуваат со себе заклучоците.

# Максимизација на несигурноста (2)

- Според овој принцип, заклучоците при амплијативното заклучување ја максимизираат релевантната несигурност во условите што ги прикажуваат претпоставките.
- Принципот гарантира дека заклучоците максимално ќе одбегнуваат директен одговор во однос на информациите што не се содржани во претпоставките.
- Кај стохастичките системи, принципот на максимизација на несигурноста се поистоветува со принципот за максимална ентропија.



# Минимизација на несигурноста

- Основна идеја: при дедуктивното расудување, да се изгубат што е можно помалку информации.
- Упростувањето се врши со примена на три стратегии:
  - ☐ елиминирање на некои ентитети од системот
  - ☐ обединување на некои ентитети од системот
  - ☐ разбивање на целиот систем во соодветни потсистеми
- Применлива е и во областа на разрешување конфликтни проблеми.

# Непроменливост на несигурноста

- Кога еден систем се трансформира, тогаш количеството информации во системот што се добива треба да бидат што е можно поблиски до оригиналот.
- При преминот:
  - ☐ количеството на несигурност сврзано со ситуацијата да се запази при преминот од теоријата  $T1$  во  $T2$
  - ☐ степенот на доверба во  $T1$  да се распредели на составните делови од  $T2$  со соодветна скала
- Принципот на непроменливост на несигурноста се вика и принцип за зачувување на информацијата.

# Практични решенија за несигурноста

- Чисто логичко расудување со игнорирање на несигурноста
- Веројатносно расудување
- Расудување според дадена вредност (Default reasoning) кое се базира врз верувањето во нешто, додека не се појави нешто во кое може повеќе да се верува
- Правила со фактор на несигурност
- Ракување со незнаењето (D-S теорија)
- Неодреденост (непрецизирана логика, матна логика – fuzzy logic)

# Врската меѓу логиката и веројатноста

- Веројатноста и логиката користат ист онтолошки посветен (commitment) пристап
- Секој исказ се третира како настан на кој му е доделен степен на верувањето (degree of belief)
- Од друга страна, на настаните им се доделува вредност на вистинитоста

# Како се дефинира веројатноста на исказите

- Нека е даден еден исказ  $A$
- Нему му се доделува веројатноста  $P(A)$
- Притоа, и тука важи условот дека
$$0 \leq P(A) \leq 1$$
- Ако исказот  $A$  е вистинит, тогаш  $P(A)=1$
- Ако исказот е лажен, тогаш  $P(A)=0$
- Исказот мора секогаш да биде или вистинит или лажен, но  $P(A)$  го претставува нашиот степен на верување дека е вистинит или лажен.

# Пример:

- Ако кажете дека утре времето ќе биде променливо облачно, во тој случај во метеоролошката прогноза ќе стои и процентуалната можност за дожд (на пр. 60%)
- Тоа значи дека веруваме дека можноста да заврне има веројатност 0,6, па затоа:  
 $P(\text{време} = \text{дожд}) = 0,6$
- Ако вредностите во доменот се:  
сонце, дожд, снег, облаци,  
тогаш важат следниве аксиоми на веројатноста:

# Основните аксиоми на веројатноста за примерот:

- $0 \leq P(\text{сонце}) \leq 1$
- $0 \leq P(\text{дожд}) \leq 1$
- $0 \leq P(\text{снег}) \leq 1$
- $0 \leq P(\text{облаци}) \leq 1$

$$P(\text{сонце}) + P(\text{дожд}) + P(\text{снег}) + P(\text{облаци}) \\ = 1$$

Ова равенство произлегува од условот дека вредностите во доменот се исцрпни и заемно исклучиви.

# Мало потсетување

- Кога се случуваат настаните:
  - $A \cup B$
  - $A \cap B$
- Кој настан се означува со  $\bar{A}$ ?
- А што значат  $B/A$  и  $A/B$ ?
- Настаните при расудувањето се логички искази, па тогаш операторите над настаните преминуваат во логички оператори.



# Премин кон расудувањето

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

$$P(A \cap B) = P(A) P(B/A)$$

$$P(A \cap B) = P(B) P(A/B)$$

$$P(\bar{A}) = 1 - P(A)$$

т.е.

$$P(A \vee B) = P(A) + P(B) - P(A \wedge B)$$

$$P(A \wedge B) = P(A) P(B/A)$$

$$P(A \wedge B) = P(B) P(A/B)$$

$$P(\neg A) = 1 - P(A)$$

# Донесување одлука во услови на несигурност

Пример од Расел и Норвиг:

- Треба да стигнете на аеродром.

Претпоставете дека верувате во следново:

$$P(A_{25} \text{ ќе ве донесе на време} \mid \dots) = 0.04$$

$$P(A_{90} \text{ ќе ве донесе на време} \mid \dots) = 0.70$$

$$P(A_{120} \text{ ќе ве донесе на време} \mid \dots) = 0.95$$

$$P(A_{1440} \text{ ќе ве донесе на време} \mid \dots) = 0.9999$$

при што  $A_t$  значи дека патникот ќе тргне кон аеродромот  $t$  минути пред полетувањето.

- Која активност ќе ја преземете, односно кое време на тргнување ќе се одбере?

# Несигурност - Uncertainty

Нека е акцијата  $A_t$  = да се тргне кон аеродромот  $t$  минути пред летот

Дали  $A_t$  ќе ме доведе на аеродромот на време?

Проблеми:

1. делумна можност за набљудување (состојба на патот, плановите на другите возачи итн)
2. шум во сензорите (сообраќајни извештаи)
3. несигурност во резултатите од акциите (дупната гума итн)
4. комплексност својствена за моделирањето и предвидувањето на сообраќајот

оттука чистиот логички приод (a purely logical approach)  
или

1. ризикува неточност (falsehood): “ $A_{25}$  ќе ме доведе таму на време”,

или

1. води до заклучоци кои се премногу слаби за донесување на одлуки (decision making):

“ $A_{25}$  ќе ме доведе таму на време ако нема сообраќајка на мостот и ако не врне и ако не се дупне некоја гума ИТН ИТН”

( $A_{1440}$  изгледа разумно дека ќе ме доведе таму на време, ама ќе треба да преспијам на аеродромот ...)

# Методи за справување со несигурностите

- Предодредена (Default) или немонотона (nonmonotonic) логика:
  - Претпоставка дека колата нема дупната гума
  - Претпоставка  $A_{25}$  врши работа освен ако не е противречно со новите сознанија
- Прашања: Кои претпоставки се разумни? Како да ги наброиме сите претпоставки (што сè може да влијае врз исходот)? Како да се справиме со противречностите?
- Веројатност (Probability)
  - Се моделира степенот на верување (belief) кај агентот
  - Според достапните сознанија (сведоштва, докази) (evidence),
  - $A_{25}$  ќе ме доведе на време на аеродромот, со веројатност 0,04

# Веројатност (Probability)

Веројатносните тврдења ги сумираат ефектите на

- мрзливост: неуспех во набројувањето на исклучоците, квалификациите итн.
- незнаење (ignorance): недостиг на релевантни факти, почетни услови итн.

Субјективна веројатност:

- Веројатностите ги поврзуваат исказите за сопствената состојба на знаењето на агентот

$$\text{пр., } P(A_{25} \mid \text{без извештаи за сообраќајки}) = 0,06$$

- Веројатностите на исказите се менуваат со нови сознанија (сведоштва, докази):

$$\text{пр., } P(A_{25} \mid \text{без извештаи за сообраќајки, 5 наутро}) = 0,15$$

# Одговорот на прашањето „Кога да тргнете кон аеродромот?“

- Зависи ...

од она што се претпочита: да се пропушти авионот или да се чека предолго.

- Теоријата на корисност (Utility theory) се користи за да го претстави и заклучи она што се претпочита (represent and infer preferences)
- Теорија на одлука (Decision theory) = теорија на веројатност + теорија на корисност

# Заедничка распределба на веројатноста

- Ако за еден домен сме нашле доволно случајни променливи со кои доменот добро се опишува, тогаш се дефинира таканаречената целосна заедничка распределба на веројатноста (full joint probability distribution FJPD).
- Со избраните случајни променливи се специфицираат целосно сите информации, па затоа FJPD доделува вредности на сите можни комбинации на случајните променливи.
- $P(V_1 = v_1, V_2 = v_2, \dots, V_n = v_n)$



# Атомни настани

- Еден настан е атомен (atomic event) ако дава целосна спецификација на светот за кој агентот е несигурен.
- Атомните настани се заемно исклучителни и исцрпно го опишуваат светот.
- На пример, две булови променливи дефинираат четири заеднички атомни настани.
- Тие се поврзани со конјункција.

# Пример за FJPD

- Светот на Коко може да се претстави преку случајните променливи: птица, лета и мала.

птица	лета	мала	веројатност
T	T	T	0,19
T	T	F	0,25
T	F	T	0,01
T	F	F	0,02
F	T	T	0,01
F	T	F	0,01
F	F	T	0,32
F	F	F	0,19

# Што дознаваме од FJPD?

- $P(\text{птица} = T) = 0,19 + 0,25 + 0,01 + 0,02 = 0,47$
- $P(\text{птица} = T, \text{лета} = T) =$   
 $P(\text{птица}, \text{лета}, \text{мала}) +$   
 $P(\text{птица}, \text{лета}, \neg \text{мала}) = 0,19 + 0,25 = 0,44$
- Според ова, одговорот на прашањето дали Коко лета би било: Веројатноста дека Коко лета е 0,44.
- Со тоа, импликацијата станува матна (fuzzy) импликација, бидејќи не е сосема сигурна.
- Колкава е веројатноста дека Коко е мал?
- А колкава е веројатноста дека не е птица?

# Условната веројатност

- Условната веројатност е клучна при расудувањето, затоа што го формализира прибирањето докази и обновувањето на веројатностите.
- Пример: Колкава е веројатноста да врне во Скопје?
  - статистички е покажано дека  $P(\text{дожд}) = 0,21$
  - оваа веројатност се вика “а приори” веројатност или безусловна веројатност
  - ако времето е веќе облачно, тогаш веројатноста за дожд се зголемува и се оценува со  $P(\text{дожд} / \text{облаци})$
  - Оваа веројатност се вика “а постериори” веројатност или веројатност која се базира на некој нов доказ (evidence).

# Условната веројатност во претходниот пример

$$\begin{aligned}
 1. \quad & P(\neg \text{птица} / \text{лета}) = \\
 & P(\neg \text{птица}, \text{лета}) / P(\text{лета}) = \\
 & (P(\neg \text{птица}, \text{лета}, \text{мала}) + P(\neg \text{птица}, \text{лета}, \\
 & \neg \text{мала})) / P(\text{лета}) = \\
 & (0,01 + 0,01) / 0,46 = 0,0435
 \end{aligned}$$

$$P(A \wedge B) = P(A) P(B/A)$$

=>

$$P(B/A) = P(A \wedge B) / P(A)$$

пти ца	лет а	мал а	веројатност
T	T	T	0,19
T	T	F	0,25
T	F	T	0,01
T	F	F	0,02
F	T	T	0,01
F	T	F	0,01
F	F	T	0,32
F	F	F	0,19

# Условната веројатност во претходниот пример

1.  $P(\neg \text{птица} / \text{лета}) =$   
 $P(\neg \text{птица}, \text{лета}) / P(\text{лета}) =$   
 $(P(\neg \text{птица}, \text{лета}, \text{мала}) + P(\neg \text{птица}, \text{лета}, \neg \text{мала})) / P(\text{лета}) =$   
 $(0,01 + 0,01) / 0,46 = 0,0435$
2. Како е пресметана маргиналната веројатност  $P(\text{лета})$ ?
3. Дали важи  $P(\text{птица} / \text{лета}) = 1 - P(\neg \text{птица} / \text{лета})$ ?

# Условна независност

- Една променлива  $V$  е условно независна од множество променливи  $V_i$  за зададено множество променливи  $V_j$  ако важи дека

$$P(V/V_i, V_j) = P(V/V_j)$$

- Оваа состојба се означува со  $I(V, V_i/V_j)$
- Една променлива  $V_i$  е независна од друга променлива  $V_j$  во множеството  $V$  ако важи дека

$$P(V_i, V_j / V) = P(V_j / V) * P(V_i / V)$$

- Променливите  $V_1, \dots, V_k$  се заемно условно независни ако

$$P(V_1, V_2, \dots, V_k / V) = \prod P(V_i / V_{i-1}, \dots, V_1, V)$$

- Ако  $V$  е празно, тогаш важи дека:

$$P(V_1, V_2, \dots, V_k) = P(V_1) * P(V_2) * \dots * P(V_k)$$

# Комбинирање повеќе докази користејќи JPD

- За одредување на веројатноста за истовремено исполнување на  $n$  услови се користи правилото на верига (Chain Rule):

$$P(A, B, C, D) = P(A/B, C, D) * P(B/C, D) * P(C/D) * P(D)$$

- Редоследот на променливите не влијае врз резултатот.



# Пример за условна веројатност која се базира на повеќе докази

$$P(\text{птица} / \text{лета, мала}) = ?$$

$$P(\text{птица} / \text{лета, мала}) =$$

$$P(\text{птица, лета, мала}) / (P(\text{лета} / \text{мала}) * P(\text{мала})) =$$

$$P(\text{птица, лета, мала}) / (P(\text{лета, мала})) =$$

$$P(\text{птица, лета, мала}) / (P(\text{птица, лета, мала}) + P(\neg \text{птица, лета, мала})) =$$

$$0,19/(0,19+0,01)=0,95$$

# Резиме

- Веројатноста е формализам кој одлично ги решава проблемите на несигурното знаење
- Заедничката дистрибуција на веројатностите ја одредува веројатноста на атомните настани.
- Одговорите на прашањата се сведуваат на сумирање на веројатностите на атомните настани.
- Независноста на настаните го намалува напорот за одредување на заедничката веројатност.
- Но, овој услов ретко кога може да биде исполнет.

# Како да се олесни пресметувањето?

- JPD е тешко да се определи, посебно кога бројот на настаните расте.
- Во голем број случаи, не може да се определи условната веројатност.
- Алтернативно решение се мрежите на Bayes.
- Тие претставуваат податочна структура преку која се одредува зависноста меѓу променливите и дава спецификација на нивното врзување.
- Мрежите на Bayes се викаат и мрежи на верувањето (Belief Networks)

# Причинско-последичен (каузален) поглед



- Непозната хипотеза H за која сакаме да направиме порценка за неизвесноста и да донесеме некоја одлука.
- Почнуваме со некое независно уверување за H (на пр., еден од секои 1000 луѓе ја има болеста).
- Ова наше верување / знаење се менува во зависност од доказниот материјал (се зголемува или намалува во зависност од веројатноста сврзана за тој доказ).

# Мотивација\*

- Користење на мамографијата како прогноза за дијагнозата
- Независната веројатност значи дека една од 100 тестирани жени има рак  
 **$P(h) = 0,01$**
- Ратата на потврдна дијагноза и лажна позитивна дијагноза се:  
 **$P(e|h) = 0,8$**  и  
 **$P(e/\text{not } h) = 0,1$**

$$\begin{aligned} P(h|e) &= \frac{P(e|h)P(h)}{P(e)} \\ &= \frac{P(e|h)P(h)}{P(e|h)P(h) + P(e|\neg h)P(\neg h)} \\ &= \frac{0.8 \times 0.01}{0.8 \times 0.01 + 0.1 \times 0.99} \\ &= \frac{0.008}{0.008 + 0.099} \\ &= \frac{0.008}{0.107} \\ &\approx 0.075 \end{aligned}$$

# Примена на правилото на Bayes\*

- Се применува често во ситуации во кои ги знаеме распределбите на зависните веројатности и сакаме да пресметаме некоја независна
- Пример: поставување медицинска дијагноза
- Лекарот знае дека менингитисот ( $m$ ) предизвикува вкочанетост на вратот ( $b$ ) во 50% случаи.
- Веројатноста пациентот да има менингитис е  $1/50000$ , а веројатноста да се има вкочанет врат е  $1/20$ .

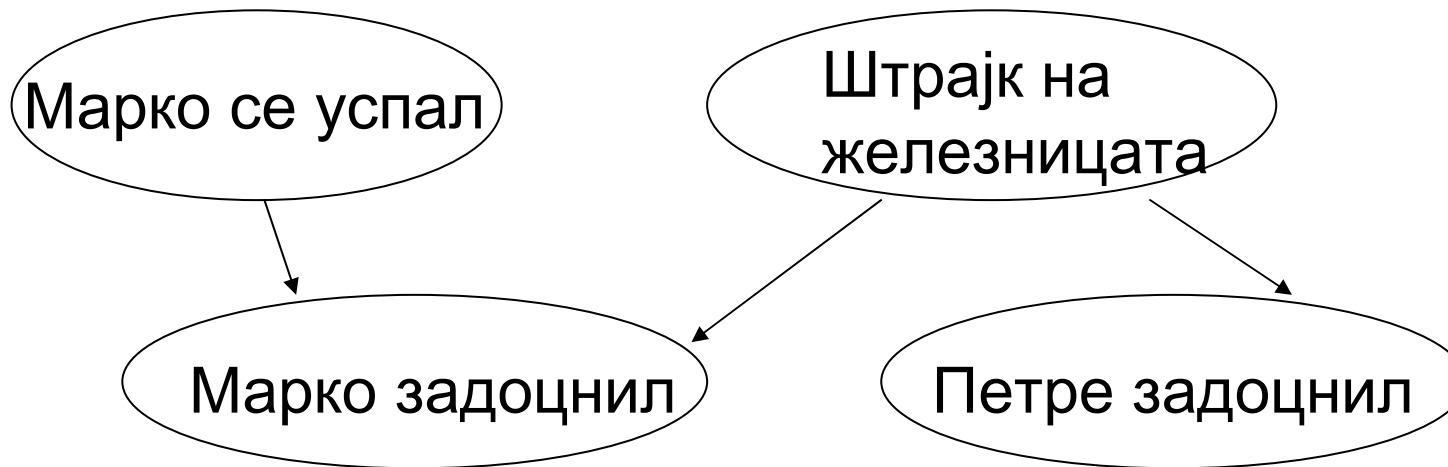
$$P(b/m) = 0,5 \quad P(m) = 1/50000 \quad P(b) = 1/20$$

$$P(m/b) = P(bm) / P(b) = P(b/m) * P(m) / P(b) = 0,0002$$

# Здружување на сведоштвото

- Ажурирањето на правилото на Bayes дозволува доказите да се вклучуваат еден по еден (нови информации, нови сведоштва).
- Со секој нов доказ, веројатноста на непознатата променлива се зголемува пропорционално со фактор зависен од новата информација.
- Системите со поголема робусност правилно ги одразуваат промените во околината.
- Независноста меѓу различните променливи ги поедноставува пресметките при користењето на правилото на Bayes (што резултира со помал број зависимости при пресметката).

# Здружување на сведоштвото



- Ако постои доказ дека Марко задоцнил, тоа значи дека уверувањето на агентот дека двете можни причини се случиле се зголемила. На пример, веројатноста на штрајк се зголемила од 0,1 на 0,13, а за успивање од 0,4 на 0,41.
- Но, ако имаме сведоштво (доказ) дека и Петре задоцнил, според правилото на Bayes веројатноста за штрајк се зголемува на 0,54, а дека Марко се успал на 0,44.



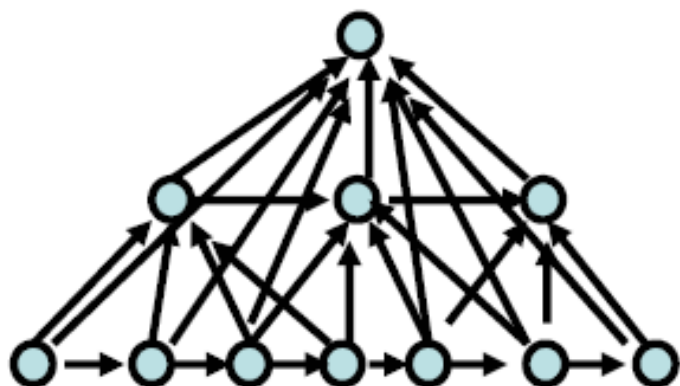
# Мрежите на Bayes

- Една мрежа на Bayes е насочен ацикличен граф чии јазли се означени со случајни променливи.
- Таа условува (stipulate) секој јазел  $V_i$  во графот да биде независен од останатото подмножество јазли.
- Оваа претпоставка е предуслов за условна независност (conditional independence)
- Ако  $A(V_i)$  е било кое множество јазли кои не се наследници на  $V_i$ , тогаш нека  $P(V_i)$  се сите следни наследници во графот.
- Во тој случај важи дека:

$$P(V_1, V_2, \dots, V_k) = \prod P(V_i / P(V_i))$$

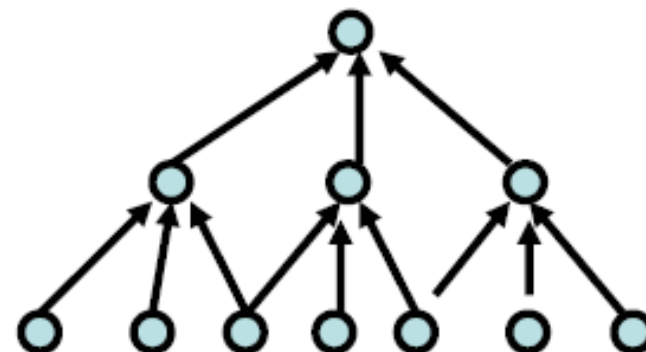
# Важноста на независноста\*

n binary nodes,  
fully connected



$2^n - 1$  independent numbers

n binary nodes  
each node max. 3 parents



less than  $2^3 \cdot n$   
independent numbers

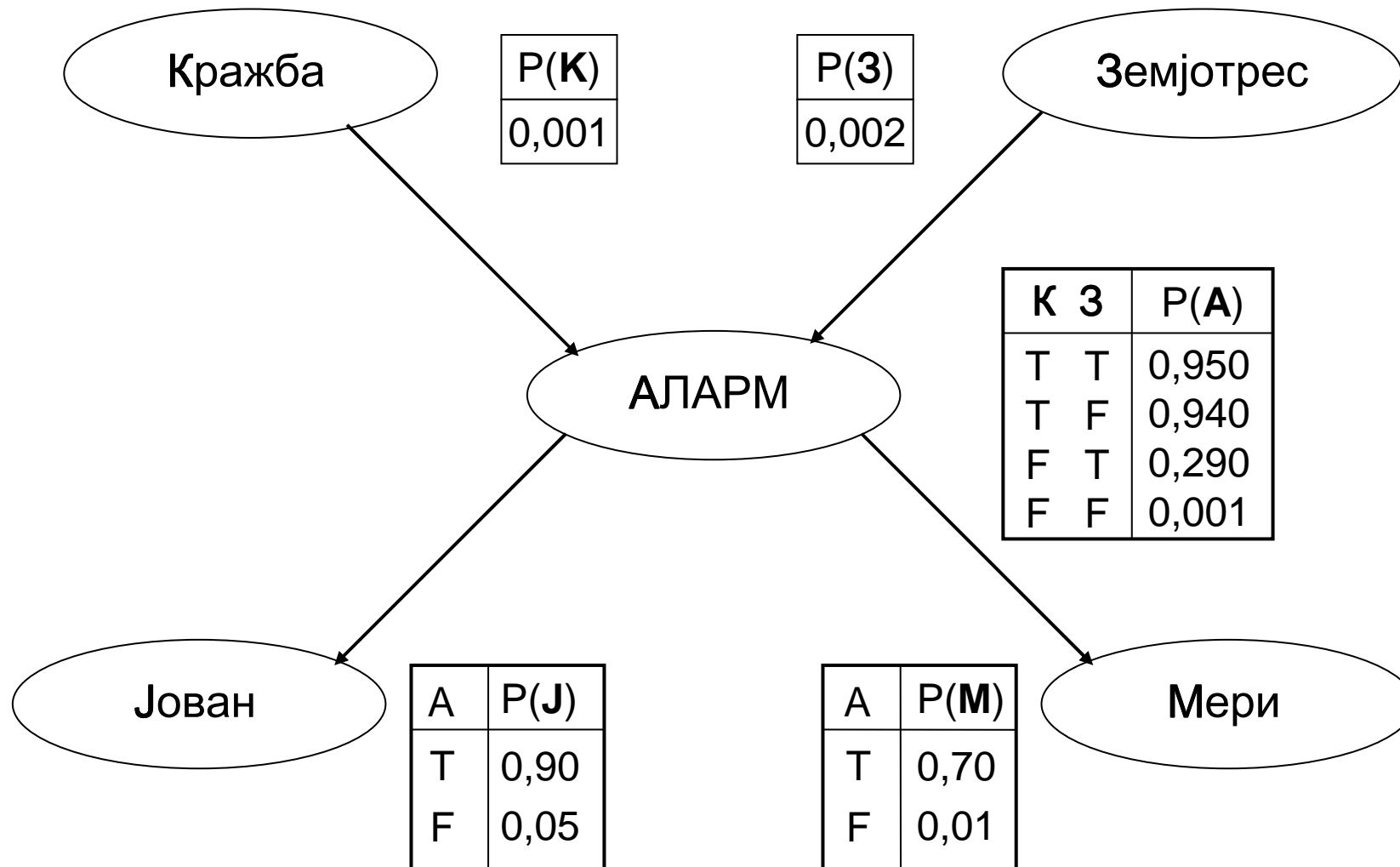
\* Сладот е преземен од <http://www.blutner.de/uncert/index.htm>

# Пример 1: Кражба или земјотрес?

- Сте набавиле нов аларм против кражба
- Тој се активира секогаш кога се врши обид за кражба, но, реагира и на мали земјотреси.
- Со соседите Јован и Мери имате договор, секогаш кога ќе го слушнат алармот, да ви се јават.
- Јован се јавува секогаш кога ќе го слушне алармот, но често го меша со телефонското звонење (се јавува и тогаш).
- Мери слуша силна музика, па понекогаш не го дослушува алармот.
- Базирајќи се врз фактот кој се јавил, а кој не, да се процени веројатноста за кражба.

- Примерот е од Judea Pearl, а преземен е од Расел и Норвиг.

# Јован се јавува, а Мери не. Дали дошол крадецот?



# Веројатноста е ...

- Алтернативниот запис на мрежите на Bayes е:

$$P(x_1, \dots, x_n) = \prod P(x_i / \text{Roditeli}(x_i))$$

- Притоа, од топологијата на мрежата зависи кои јазли се родителски.
- Ако не се знае дали Јован навистина го слушнал алармот тогаш:

$$P(J, \neg M, A, K, \neg 3) + P(J, \neg M, \neg A, K, \neg 3) = 0,00025625646$$

## Детално:

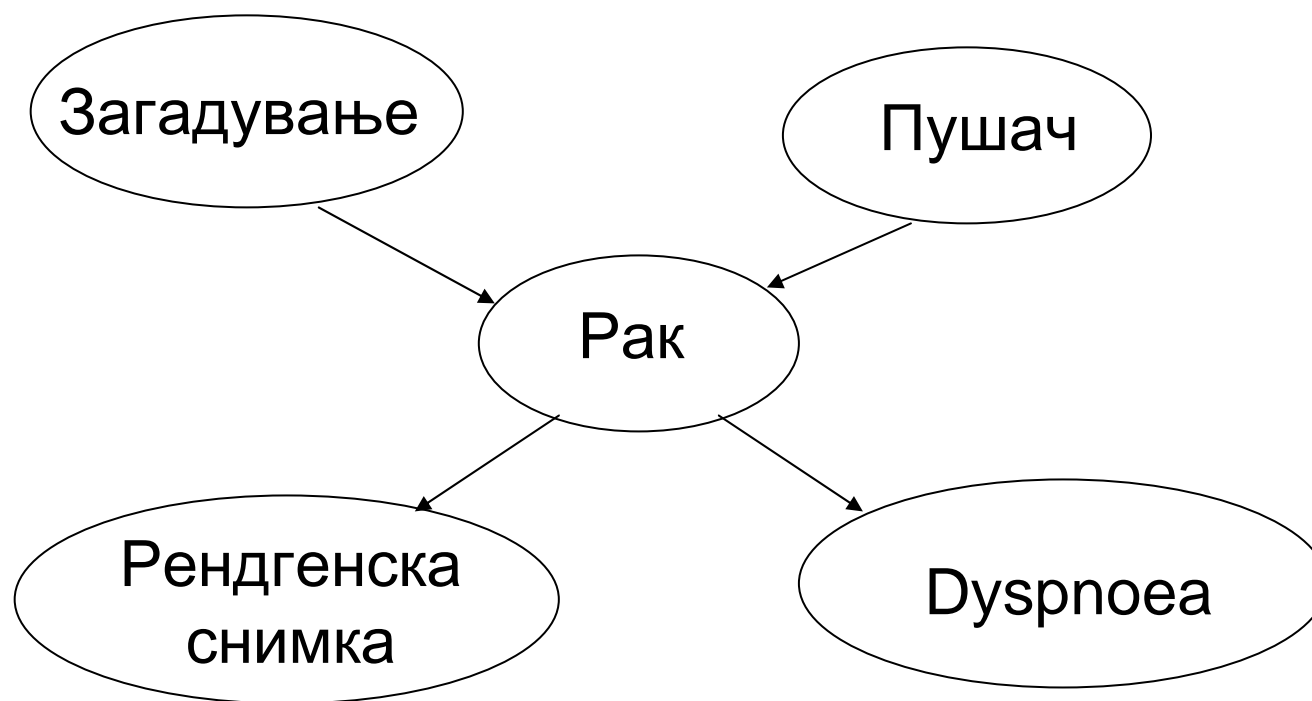
$$\begin{aligned} \blacksquare P(J, \neg M, A, K, \neg 3) &= \\ P(J/A) * P(\neg M/A) * P(A/K, \neg 3) * P(K) * P(\neg 3) &= \\ 0,90 * 0,30 * 0,940 * 0,001 * 0,998 &= \\ 0,0002532924 \end{aligned}$$

$$\begin{aligned} \blacksquare P(J, \neg M, \neg A, K, \neg 3) &= \\ P(J/\neg A) * P(\neg M/\neg A) * P(\neg A/K, \neg 3) * P(K) * & \\ P(\neg 3) &= \\ 0,05 * 0,99 * 0,060 * 0,001 * 0,998 &= \\ 0,00000296406 \end{aligned}$$

# Пример 2 за Бејесова мрежа

- Пациентот чувствува тешкотии во движењето (Dyspnoea) и исплашен дека можеби има рак на белите дробови оди на лекар.
- Лекарот знае дека овој симптом може да биде последица и на други заболувања, како на пример, туберкулоза (ТВ) или воспаление на белите дробови.
- Тој исто така знае дека постојат други релевантни информации кои треба да се земат предвид, како што е податокот дали пациентот е пушач и какво е загадувањето на средината во која живее.
- Рендгенската снимка на белите дробови може да индицира постоење на ТВ или на рак на белите дробови.

# Илустрација на примерот

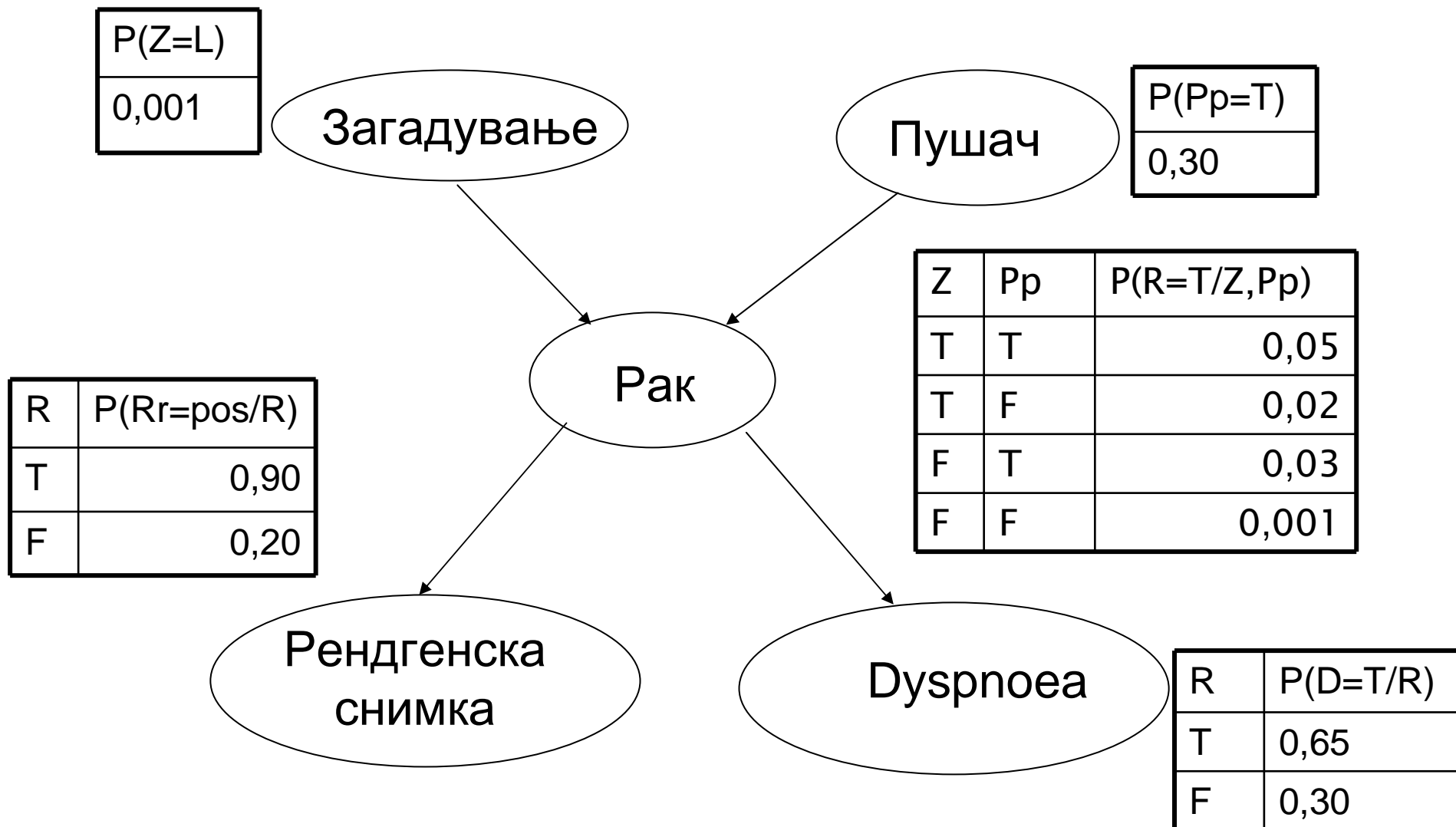




# Фактори вклучени во Бејесовата мрежа

Node name	Type	Values
<i>Pollution</i>	Binary	$\{low, high\}$
<i>Smoker</i>	Boolean	$\{T, F\}$
<i>Cancer</i>	Boolean	$\{T, F\}$
<i>Dyspnoea</i>	Boolean	$\{T, F\}$
<i>XRay</i>	Binary	$\{pos, neg\}$

# Продолжение на примерот



# Како се градат мрежите на Bayes

- Се одбира множество случајни променливи кои го опишуваат проблемот
- Случајните променливи се означуваат како  $X_1, \dots, X_n$
- Од  $i=1$  до  $n$ 
  - Отвори нов јазел за  $X_i$  во мрежата
  - Множеството родители на  $X_i$  е најмалото можно множество јазли кои веќе се во мрежата, а се условно независни со  $X_i$ . Тие стануваат  $Roditeli(X_i)$
  - Воспостави лак од секој  $Roditeli(X_i)$  кон  $X_i$
  - Ако  $X_i$  има барем еден родител, тогаш дефинирај ја табелата на условната веројатност. Во спротивно, дефинирај ја веројатноста на јазелот.

# Практичен начин за конструкција на мрежата

- Најдобро е да се започне со јазелот / јазлите кои се причина, а потоа се додаваат јазлите врз кои тие делуваат итн., сè додека не се дојде до лисјата во графот (каузална / причинско-последична структура)
- Каузалниот редослед овозможува директна репрезентација на условните веројатности, бидејќи секој друг редослед ќе бара внесување нови врски во мрежата за воспоставување на независност.
- Ако се почне со други јазли или се користи друг редослед се добиваат посложени структури кои бараат повеќе пресметки / веројатности (колку што има заедничката распределба).

# Сведоштво - практично

- *Конкретен доказ* (пример, дека пациентот е пушач)
- *Негативен доказ* (пример, дадена променлива не е во дадена состојба, што значи дека е во некоја од преостанатите)
- *Несигурен или можеен доказ* кога изворот на информации не е сигурен за тоа (пример, радиологот е 80% сигурен дека рендгенската снимка е позитивна)
- Расудувањето со помош на Бејесовите мрежи е флексибилно: се додава ново сведоштво за било кој јазел (информации за околината), се ажурираат уверувањата за сите други.

# Расудувањето во мрежите на Bayes

- Да се најде условната (posterior) веројатност за одредени променливи - хипотези врз основа на вредностите кои ги имаме како доказ (evidence variables)
- Агентот ги добива веројатностите за доказните променливи врз основа на своите перцепции или слични расудувања и поставува прашање за можните веројатности на другите променливи, за да може да одлучи која акција да ја одбере.

# Резиме за мрежите на Bayes

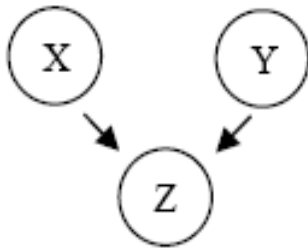
- Се создаваат интуитивно, но притоа, ако од јазелот  $X$  води стрелка кон  $Y$ , во тој случај,  $X$  има директно влијание врз  $Y$ .
- Се викаат и причинско-последични мрежи (causal networks) во кои лаците ги претставуваат директните причини.
- Пресметките се вршат со помош на табели на условната веројатност (conditional probability tables CPTs).
- Услови:
  - ☐ настаните се независни
  - ☐ треба да се запази редоследот причина – последица

# D-одделување (D-Separation)\*

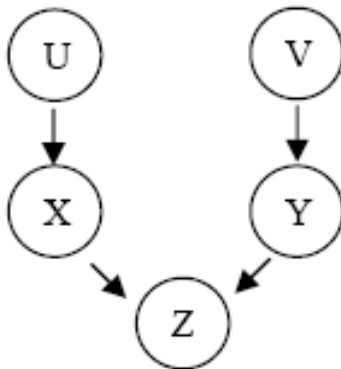
- Мрежите на Bayes може да се градат само ако секој јазел е условно независен од сите јазли што не се негови наследници
- Ова е услов на локалната семантика.
- D-одделувањето е критериум кој е построг од локалната семантика.



# Илустрација на двата критериума\*



Гледано локално, X и Y се независни од претходните јазли.

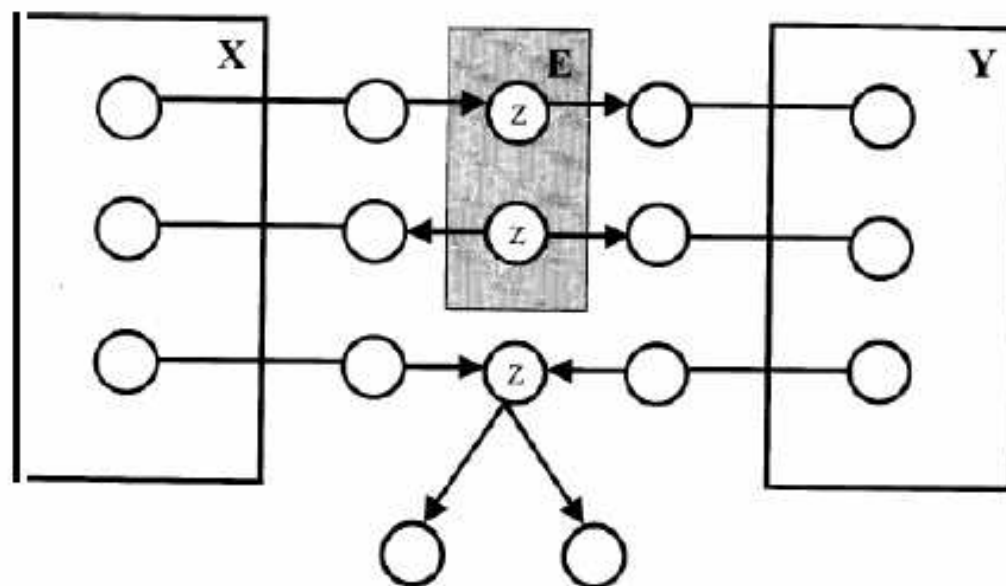


X и Y веќе не се независни од претходните јазли, но сепак се меѓусебно независни.

Како да се одреди дали едно множество јазли  $X$  е независно од друго множество  $Y$ , во услов на множество докази  $E$ ?\*

- Ако секој ненасочен пат од јазел во  $X$  кон јазел во  $Y$  е одделен со  $E$ , тогаш  $X$  и  $Y$  се условно независни преку  $E$ .
- Множеството јазли  $E$  ги одделува множествата јазли од  $X$  и  $Y$  ако секој ненасочен пат од јазелот од  $X$  кон јазел од  $Y$  е блокиран од  $E$ .
- Патот е блокиран со множеството јазли од  $E$  ако постои јазел  $Z$  на патот, за кој важи:
  1.  $Z$  е во  $E$  и  $Z$  има стрелка која води кон неа и стрелка која води надвор од неа (ланец)
  2.  $Z$  е во  $E$  и двете стрелки на  $Z$  водат надвор од неа (заедничка причина)
  3. Ниту  $Z$ , ниту било кој наследник на  $Z$  не е во  $E$  и двете стрелки водат кон  $Z$  (заеднички ефект)

# Илустрација на критериумите за блокирање на патот\*



E d-separates X and Y

\* Сладот е преземен од <http://www.blutner.de/uncert/index.htm>

# Теоријата на Демпстер и Шефер (Dempster-Shafer Theory)

- Според Шефер, во теоријата на доказите се дефинираат две мерки:
  - мерка за верување (belief measure):  $Bel$
  - мерка за веродостојност (plausibility measure):  $Pl$
- Според Демпстер, овие функции се функции на верување и тие се наменети за пресликувања со повеќе вредности.

# Теоријата на Демпстер и Шефер (Dempster-Shafer Theory)

- Теоријата се базира врз две идеи:
  - Да се одреди мерка на верување (degree of belief) за дадено прашање врз основа на субјективна веројатност за сродно прашање
  - Комбинирање на мерките на верување кога тие се засновани на независни докази
- Мерката за верување може, но не мора да ги има својствата на веројатноста
- Мерката за верување се вика и функција на верување  $Bel(x)$

# Пример 1

- Фрлање на паричка кое ви го предлага сомнителен тип кој ви вели дека можете да заработите пари (не знаете дали не си поиграл со паричките)
  - Бидејќи немате доказ, степенот на вашето верување е исто и за двата одговора петка и глава:  $Bel(\text{Глава}) = 0,5$ ;  $Bel(\neg\text{Глава}) = 0,5$
  - Ако дојде експерт за парички, можно е да каже со 90% сигурност дека паричките не се менувани:  $Bel(\text{Глава}) = 0,9 \times 0,5 = 0,45$
- Теоријата не дава одговор на секое прашање

## Пример 2

- Имате субјективен впечаток за тоа дали Бети ја зборува вистината или не
  - 0,9 дека ја зборува вистината и 0,1 дека не
  - Ако дојде и ви каже дека некој ја удрил вашата кола, тоа што и верувате само 90% дека ја кажува вистината ќе ви значи дека веројатноста некој навистина да ве удрил е 0,9. Но верувањето дека никој не ви ја удрил колата е 0 (не е 0,1, бидејќи немаме доказ). Тоа нема исто значење како веројатноста 0, дека никој не ја удрил колата.
  - Вредностите на функцијата на верување се 0,9 и 0 соодветно.

## Пример 2, проширување

- Имате субјективен впечаток и за тоа дали Марија ја кажува вистината
  - 0,9 дека ја зборува вистината и 0,1 дека не
  - Ако дојде Марија и таа исто така ви каже дека некој ви ја удрил колата, веројатноста за верување на двете е  $0,9 \cdot 0,9 = 0,81$ ; веројатноста дека не може да им се верува е  $0,1 \cdot 0,1 = 0,01$ , а веројатноста дека и веруваме барем на едната е  $1 - 0,01 = 0,99$  (вашето верување дека некој ви ја удрил колата, бидејќи и двете во кажење е 0,99).
- Имаме уверување за едно прашање (дали некој ви ја удрил колата?) врз основа на субјективната веројатност за друго (колку им веруваме?)



# Интерпретација

- Интерпретација 1: се дефинира праг на веројатности
  - Пред да дојде експертот:  $Bel(\text{Глава}) = [0, 1]$  по доаѓањето:  $Bel(\text{Глава}) = [0.45, 0.55]$ .
  - Опсегот на интервалот посочува дали ви се потребни повеќе докази
- Долниот праг е мерка за верување, а горниот се вика мерка за веродостојност (plausability)

# Непрецизирана (fuzzy) претстава (логика)

- Прецизно дефинирано множество  
(crisp)  
 $A = \{a, b, c, d, e\}$
- Се дефинира функција на припадност  
(membership function)  
 $M = \{1, 1, 1, 1, 1\}$

Замислете кога функцијата на припадност би можела да  
прима вредности во интервалот  $[0, 1]$

# Fuzzy = матно, замаглено

- Fuzzy  $\neq$  фази, оти фази е множина од фаза

Значи:

- Fuzzy Logic = матна логика,
- Fuzzy concepts = матни, заматени или замаглени концепти,
- Fuzzification = заматување, замаглување
- Defuzzification = одмаглување

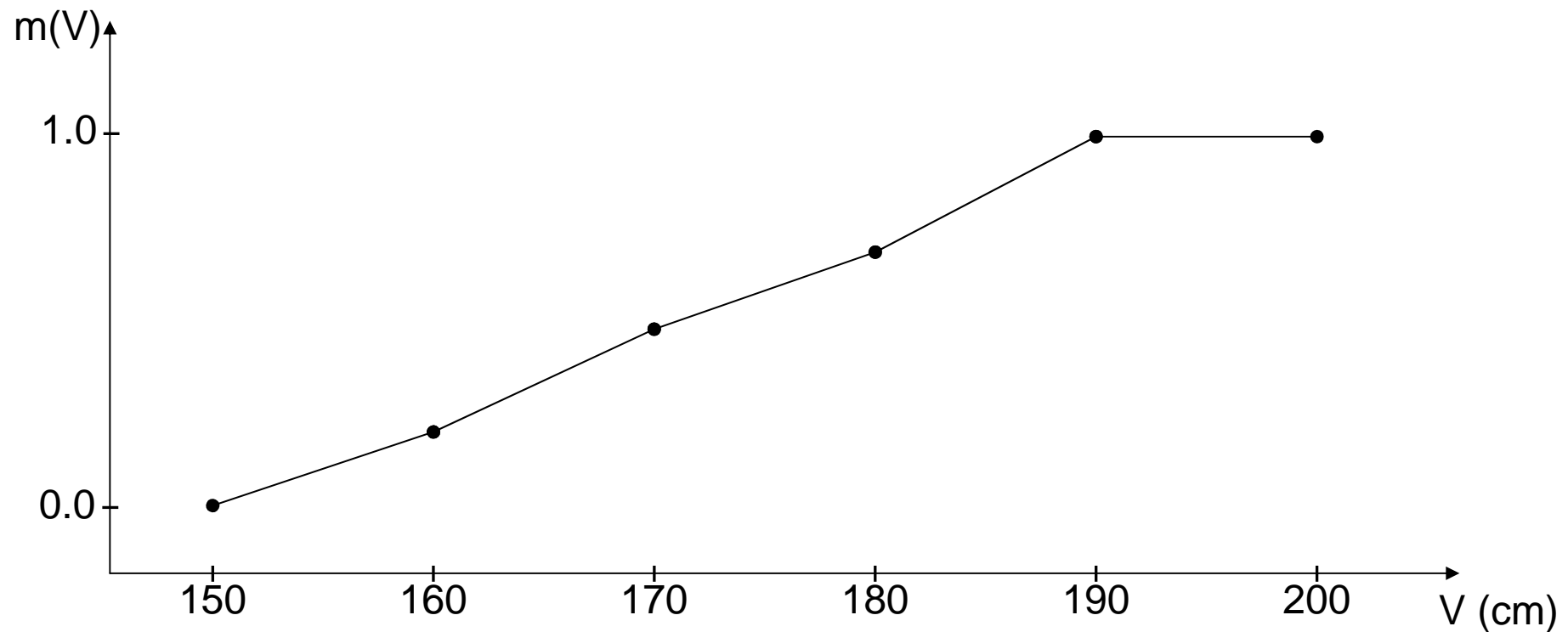
# Непрецизирани множества

- $M = \{0.7, 0.6, 0.2, 1.0\}$  на пример да биде множество на припадност на дадени елементи кон некое множество
- Па вака може да се дефинираат и некои поими (концепти)
  - Пример: Висок. Што е тоа висок човек?
    - $V = \{150, 160, 170, 180, 190, 200\}$   
 $m(V) = \{0.0, 0.2, 0.5, 0.7, 1.0, 1.0\}$



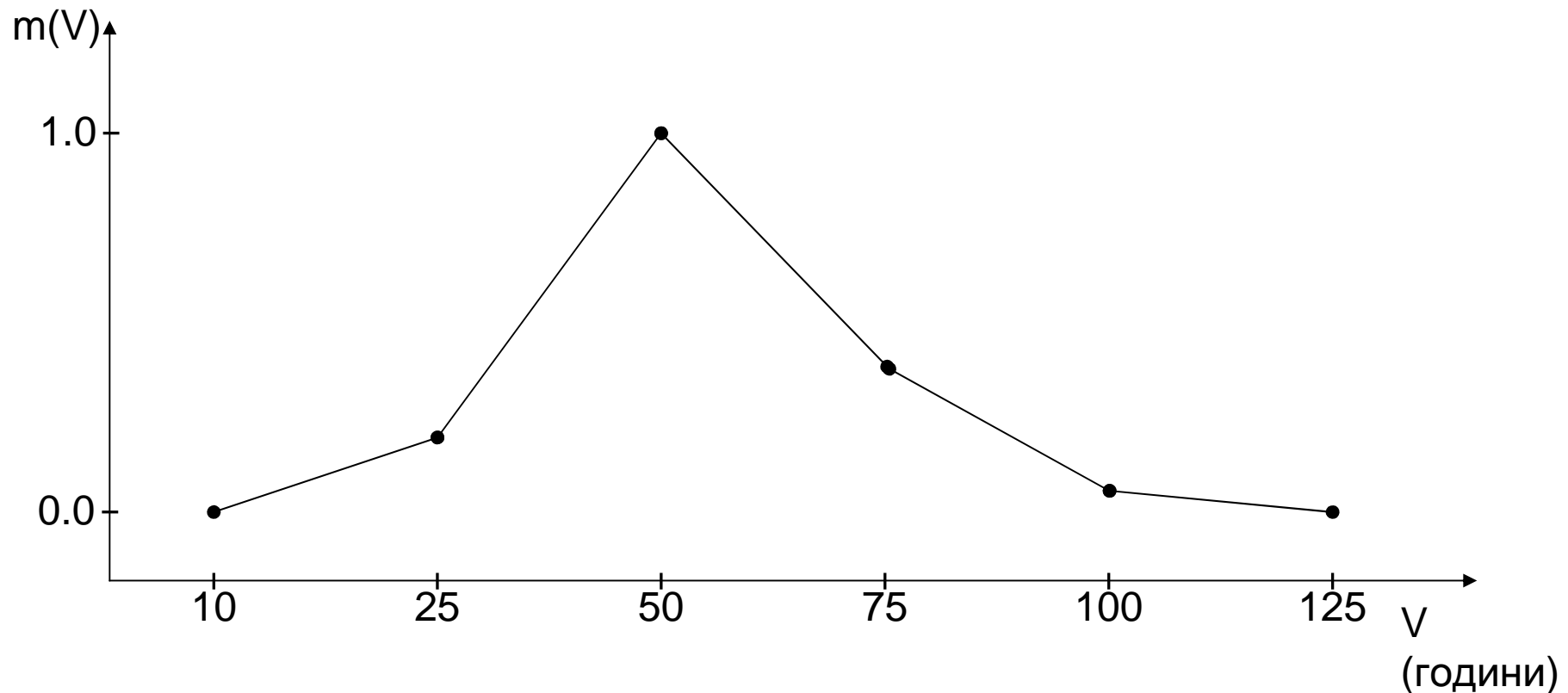
# ПОИМОТ ВИСОК ЧОВЕК

- $V = \{150, 160, 170, 180, 190, 200\}$   
 $m(V) = \{0.0, 0.2, 0.5, 0.7, 1.0, 1.0\}$



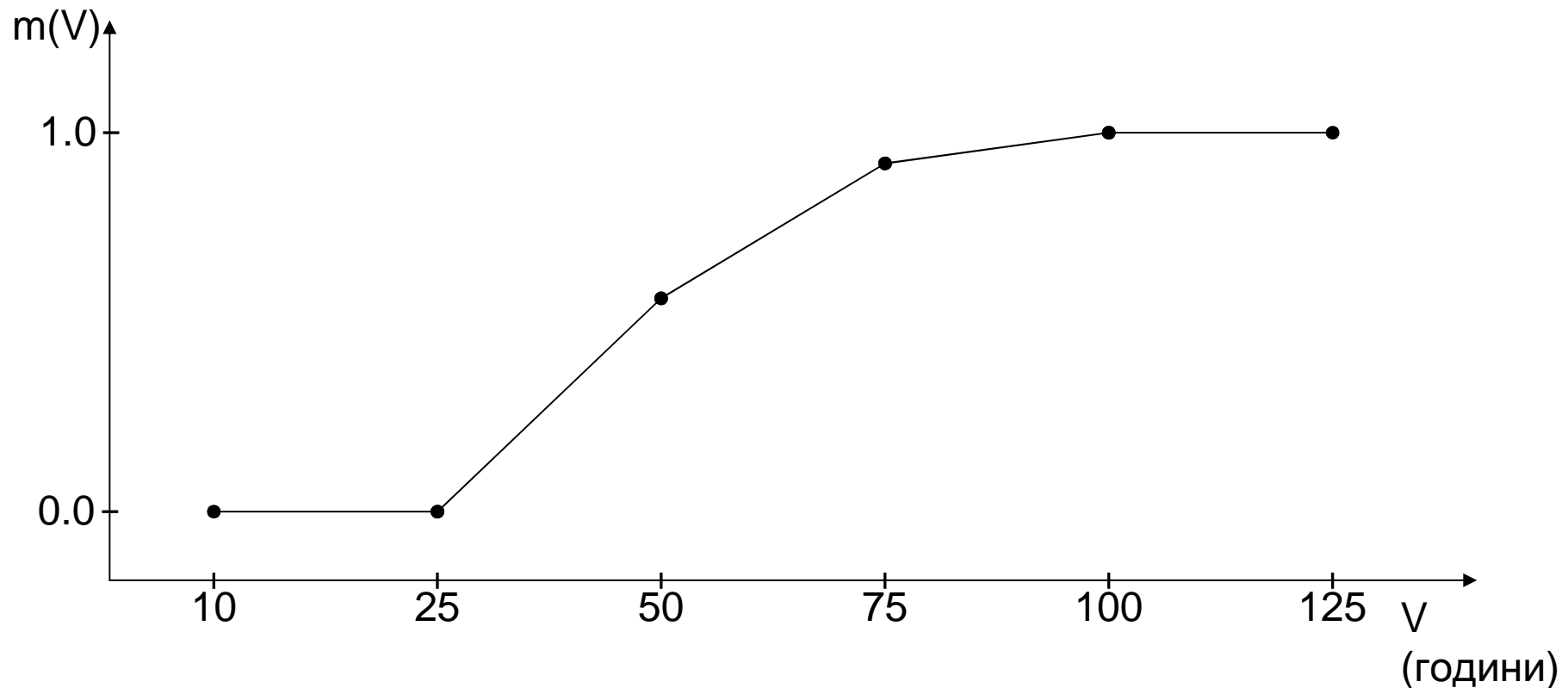
# Поимот средовечен човек

- $V = \{10, 25, 50, 75, 100, 125\}$   
 $m(V) = \{0.0, 0.2, 1.0, 0.3, 0.1, 0.0\}$



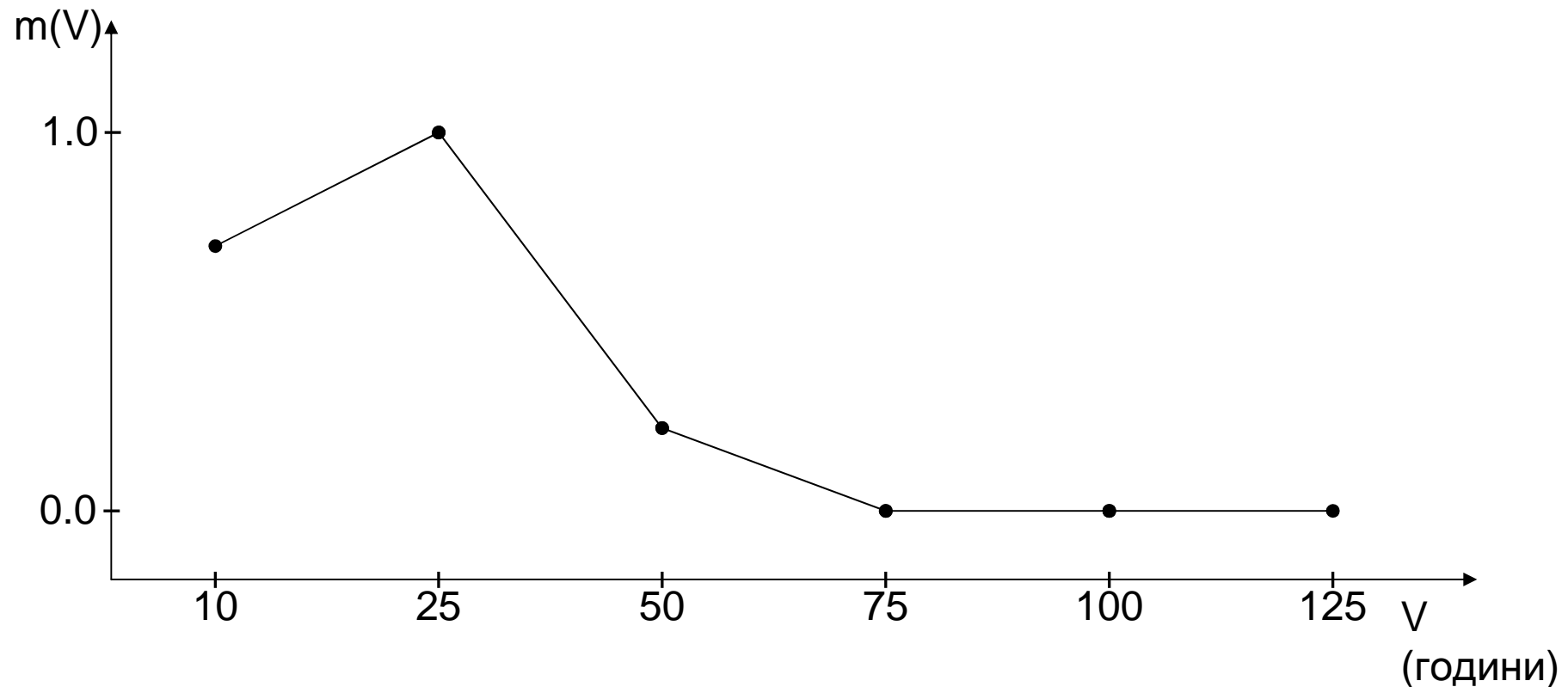
# Поимот стар човек

- $V = \{10, 25, 50, 75, 100, 125\}$   
 $m(V) = \{0.0, 0.0, 0.5, 0.9, 1.0, 1.0\}$



# Поимот млад човек

- $V = \{10, 25, 50, 75, 100, 125\}$   
 $m(V) = \{0.7, 1.0, 0.2, 0.0, 0.0, 0.0\}$





# Користена литература

- Artificial Intelligence, A Modern Approach  
2nd edition, Russel and Norvig
- Artificial Intelligence, A New Synthesis, Nils J.  
Nilsson



# Прашања?



ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Агенти способни да учат (машинско учење)

# Машинско учење

- Целта на машинското учење е вградувањето на способностите за учење кај сметачите.
- Предмет на машинското учење се студиите и компјутерското моделирање на процесите на учење и нивните манифестации.
- Машинското учење се занимава со создавањето и развојот на алгоритмите и техниките кои му овозможуваат на сметачот да учи.

# Потреба од машинското учење

- Развивање системи кои во случај рачно да се развиваат би биле тешки или скапи, а притоа бараат специфични знаења или вештини.
- Развивање системи кои автоматски се приспособуваат и прилагодуваат (adapt and customize), како на пример:
  - Персонализирани филтри за вести или пораки
  - Персонализирано подучување
- Откривање нови знаења од големи податочни бази, како на пример:
  - Анализа на пазарот
  - Текстуално рударење на медицински податоци

# Што е учење?

- Промени во даден систем кои водат кон тоа при следната појава на слична задача, системот да ја извршува поефикасно и поефективно
- Herbert Simon: “Учење е било кој процес преку кој еден систем го подобрува своето однесување врз основа на стекнатото искуство.”
- Од тривијално меморирање на факти ( $2+2=4$ ) до развивање на потполно нови теории (откривање нови знаења)

# Учењето опфаќа

- Усвојување ново декларативно знаење
- Развивање моторни и спознајни вештини по пат на инструкции или пракса
- Организација на новото знаење во општи, ефективни репрезентации
- Откривање нови факти и теории по пат на набљудувања или експерименти.

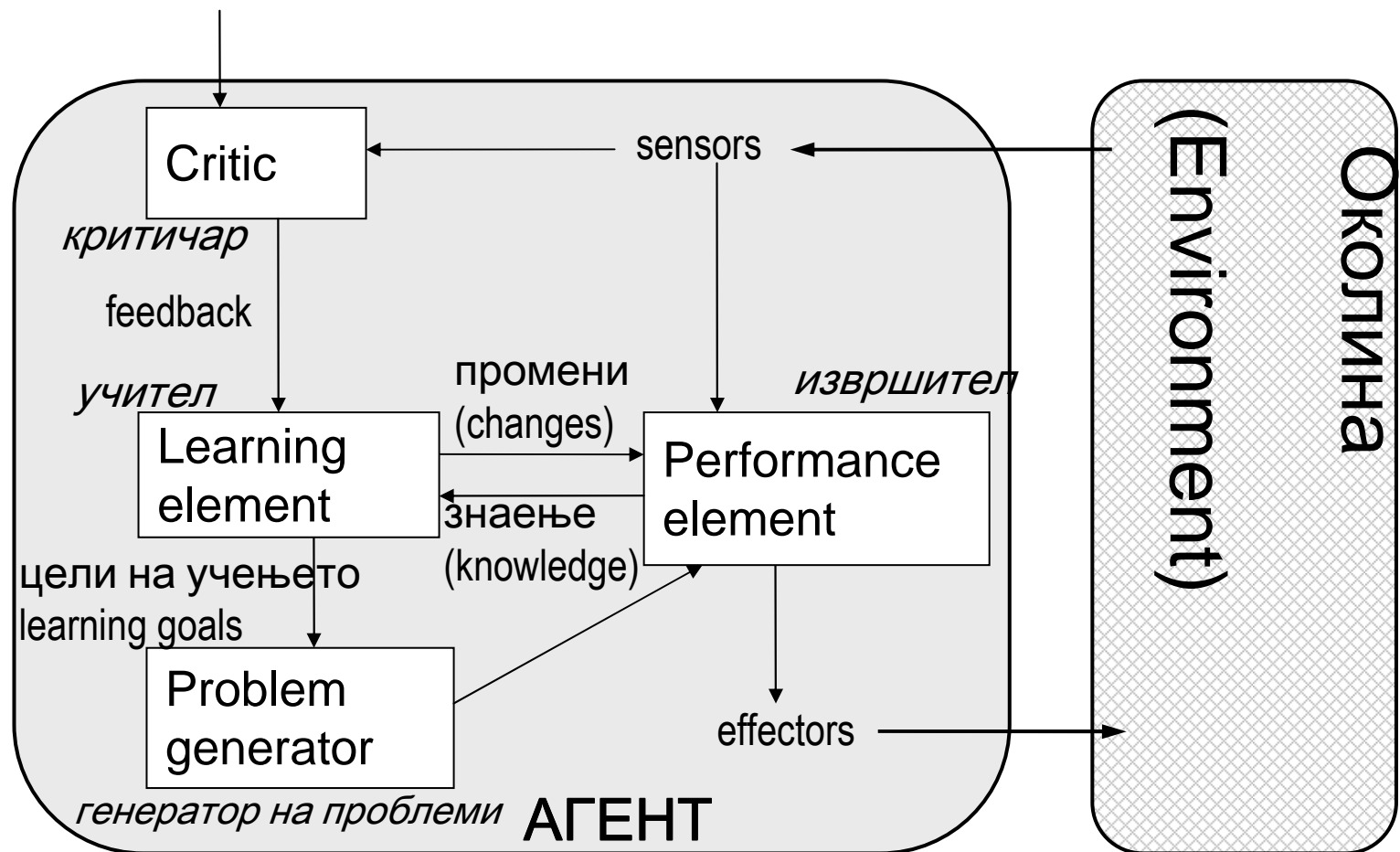
# Како учи еден агент?

- Со механичко учење или бубање (rote learning)
- Учење според инструкции (learning from instruction)
- Учење по аналогија (learning by analogy)
- Учење од набљудување и откривање (learning from observation and discovery)
- Учење од примери (learning from examples)



# Општ модел на агент кој учи

надворешен стандард за изведбата  
(external performance standard)



# Учител (Learning element)

- изајнот на учителот зависи од:
  - *КОМПОНЕНТИТЕ* на извршителот кои треба да се научат;
  - *повратната информација* која е на располагање за да се научат овие компоненти;
  - која е *репрезентацијата* на овие компоненти;

# Видови учење врз основа на повратната информација

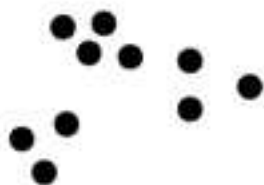
- Учење со учител – надгледувано (supervised learning): со секој примерок се проследува и точниот одговор;
- Учење без учител – самоучење, ненадгледувано (unsupervised learning): не се проследува точниот одговор;
- Учење со поттикнување (reinforcement learning) (наметнато учење): повремени награди и/или казни.

# Надгледувано учење

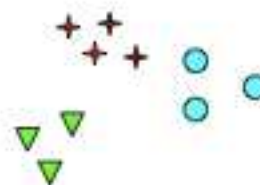
- Надгледуваното учење е техника со која се создаваат функции само врз основа на множество за обука (training set).
- Целта на ова учење е да ја предвиди вредноста на функцијата за секој влезен објект откако ќе научи како да ја предвиди вредноста на функцијата по прикажувањето примери за обука (training examples)
- Значи, станува збор за примена на обопштувањето.
- Пример: ако на системот му се претстават илјадници именки во еднина и множина, тој се обидува да најде правила за формирање на множината од еднината (споредете со слајд 55).

# Ненадгледувано учење

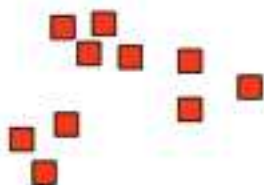
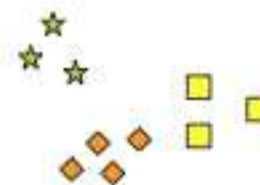
- Ненадгледуваното учење се занимава со класификација или дефинирање кластери во кои припааат влезовите.
- Тоа проучува како да се претстават влезовите за да ја одразат статистичката структура на сите можни збирки од влезни обележја.
- Пронајдените објекти кои припааат на даден кластер се слични меѓу себе (или во некаква релација), но различни од објектите кои припааат на други групи.
- Примена:
  - а се воочи некоја зависност / поврзаност (пр. групирање на слични документи на Интернет)
  - Сумаризација т.е да се намали големината на множеството податоци



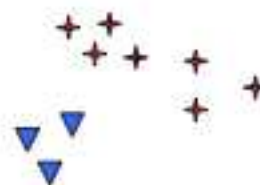
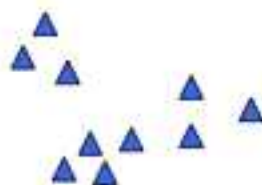
Initial points.



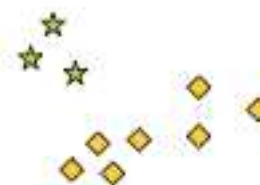
Six Clusters



Two Clusters



Four Clusters



ројот и големината на кластерите може да се одреди според неколку параметри

# Учење со поттикнување

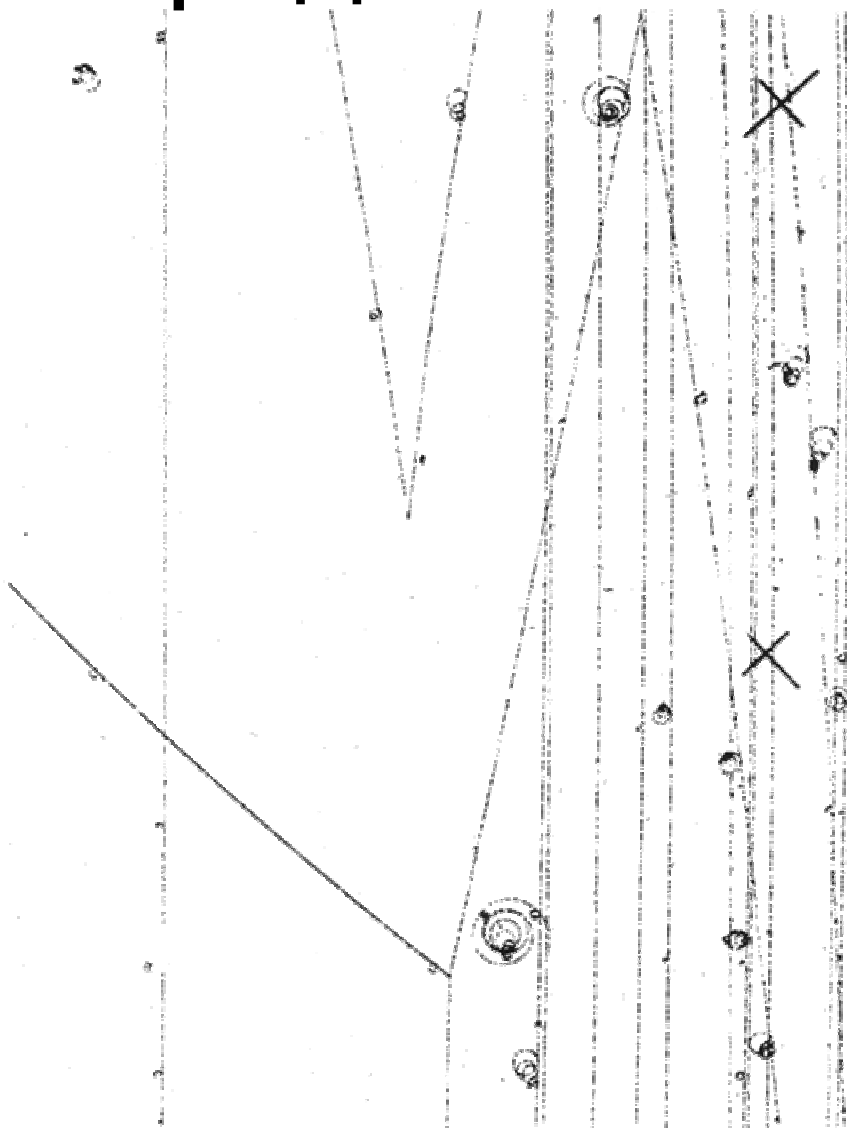
- Учењето со поттикнување се занимава со тоа кои активности треба да ги преземе еден автономен агент во околината во која делува за тоа да биде оптимално.
- Се базира на доделување награди за оние активности што доведуваат до подобро однесување, и се стреми кон тоа да ги максимизира наградите.
- Исто така може да се доделуваат и казни за оние активности кои водат до полошо однесување и се стреми кон тоа да ги минимизира казните.
- Отсуство на награда исто така може да се смета како еден вид на казна.
- Алгоритмите што тука се применуваат прават обид да најдат т.н. политики (полиси) кои оптимално ги пресликуваат состојбите на светот во активности на агентот.

# Учење со поттикнување

- Кај игрите како што е шахот, поттикот се добива дури на крајот на играта. Кај други околина, поттиците се почести.
  - Во пинг-понг, секој постигнат поен може да се смета за награда. Кога учиме како да ползиме, секое движење нанадпред е постигнување.
- Недостатокот на бакшиш на крајот на патувањето или големата сметка за поправка заради можен судир со колата пред него, му дава на агентот (таксистот) некаков показател дека таквото однесување е непосакувано.



# Предзнаење



- Мноштвото на ML истражувања во AI се концентрираат на обука на агент кој на почетокот нема никакво знаење за тоа што се обидува да научи. Тој има пристап само до примероци презентирани од неговото искуство.
- Во реалноста најголем дел од учењето кај човекот се изведува во контекст на одредено предзнаење.
- Bubble-chamber photo. Како ја гледа
  - уметник (слика од апстрактен експресионист)
  - нуклеарен физичар (постоење на нова честица со одредена маса и полнеж)

# Некои аспекти поврзани со процесот на учење

Парови спротивставени поими при учењето:

- индукција наспроти дедукција
- генерализација наспроти спецификација
- апстракција наспроти конкретизација
- конзистентност наспроти комплетност

# Индуктивно учење (1)

Кај надгледуваното учење на учителот му се проследуваат (приближно) точни вредности на функцијата за одредени влезови, и учителот се обидува да ја измени репрезентацијата на функцијата со цел нејзината вредност да се совпадне со доставената повратна информација.

$f$  е целната функција

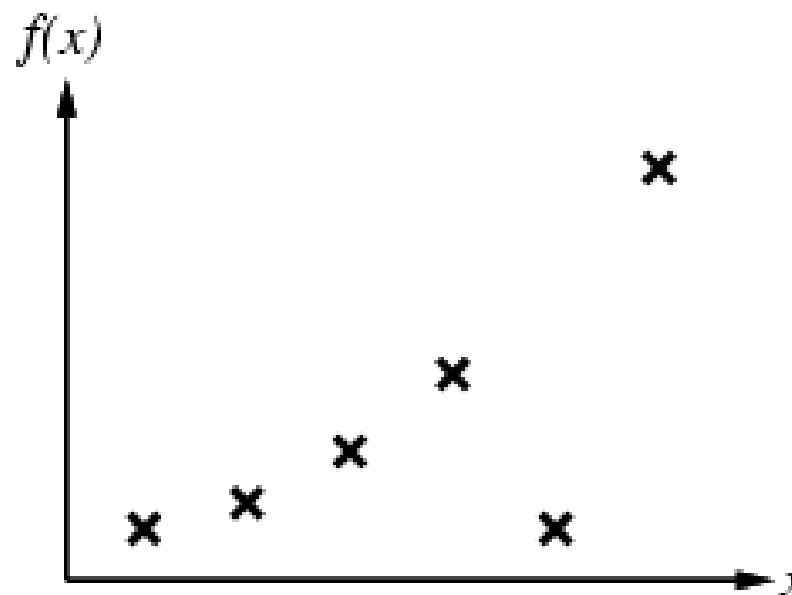
примерок е парот  $(x, f(x))$

каде  $x$  е влезот, а  $f(x)$  е вредноста на функцијата  $f$  за влезот  $x$

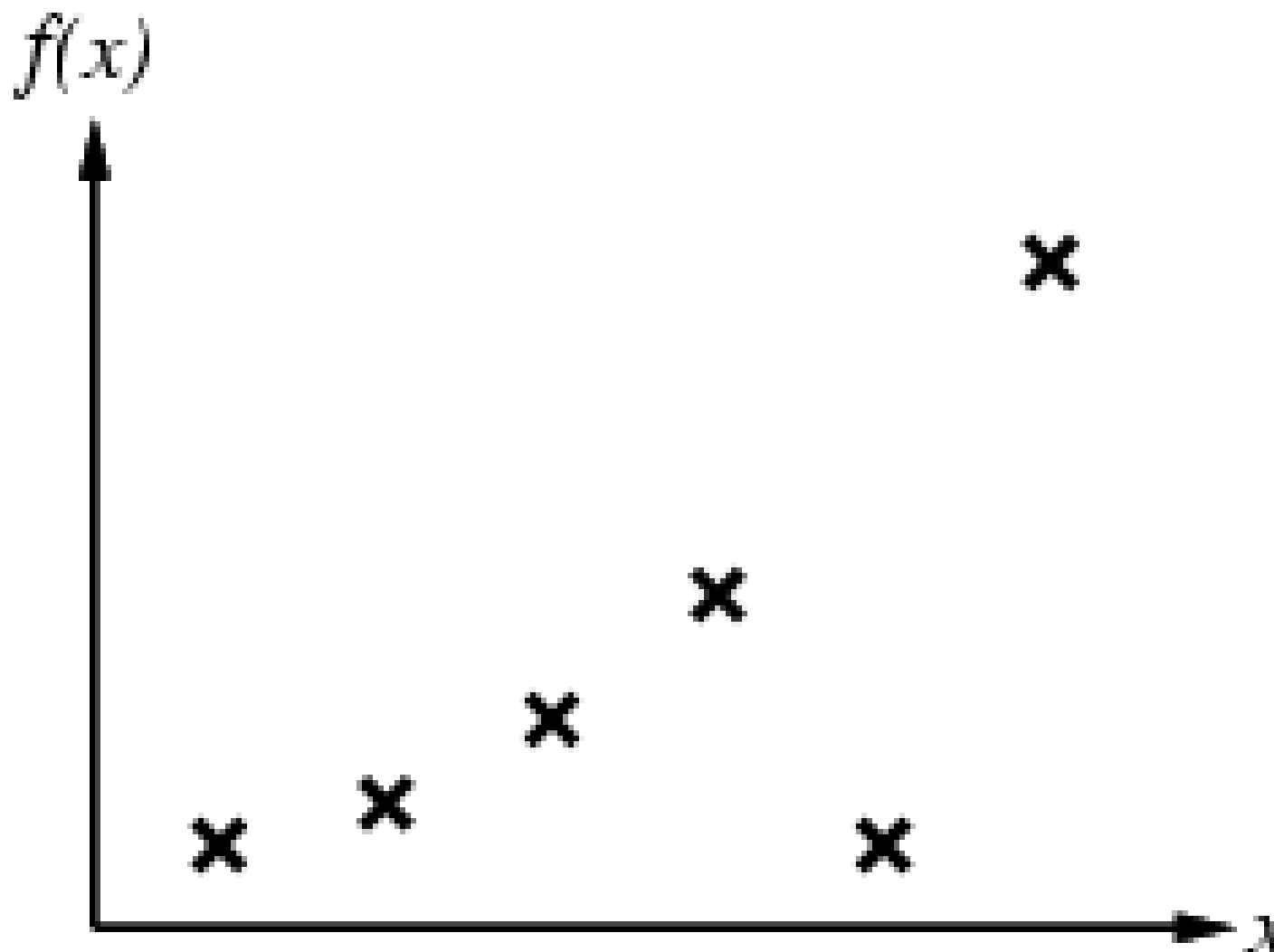
Проблем: да се пронајде хипотеза  $h$  така да  $h \approx f$ , за дадено множество примероци за обука

# Индуктивно учење (2)

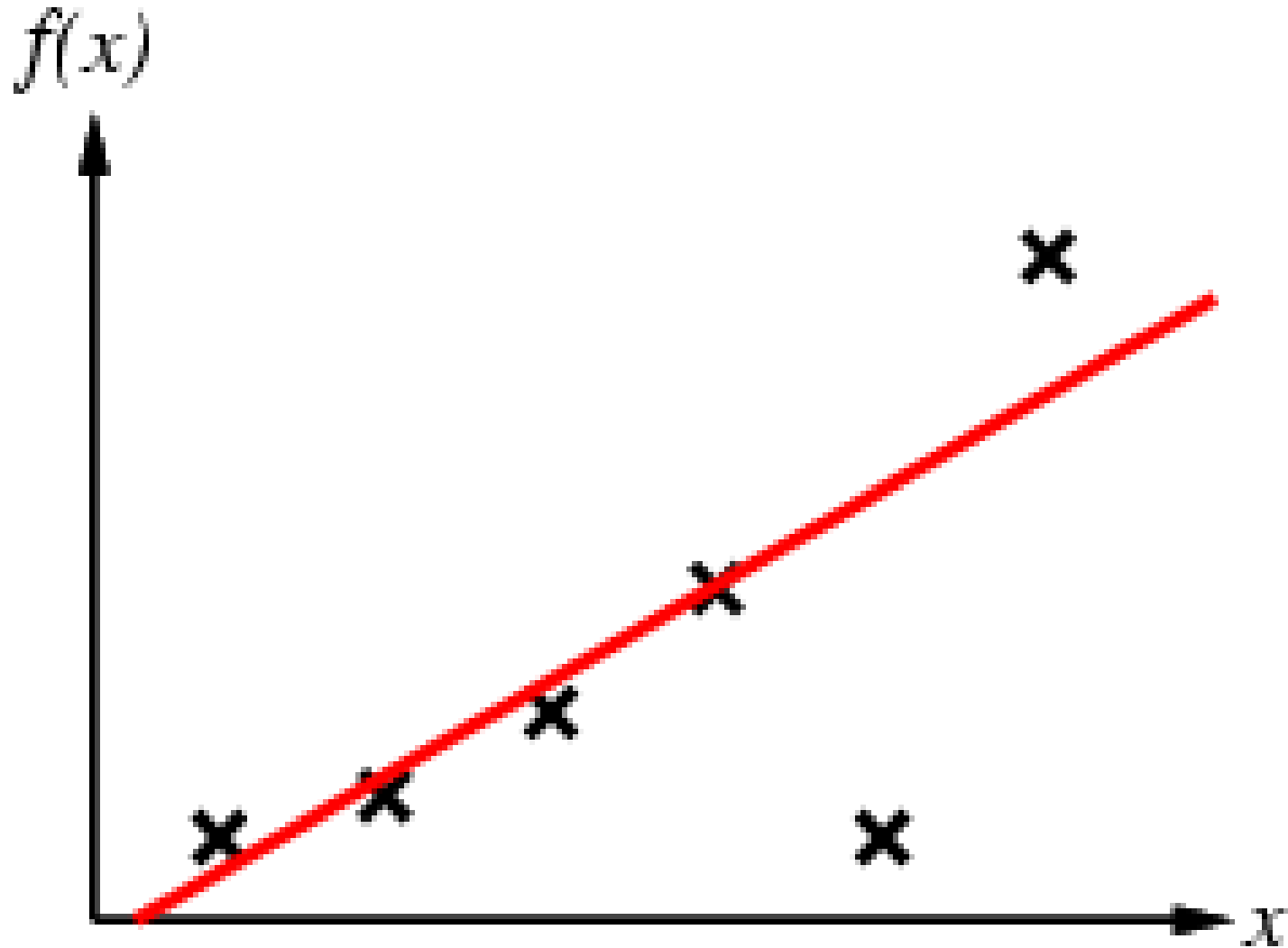
- Пронајди / нагоди функција  $h$  која ќе се совпа а со  $f$  на примероците од множеството за обука.
- $h$  е конзистентна доколку се совпа а со  $f$  за сите примероци
- Пр. curve fitting.



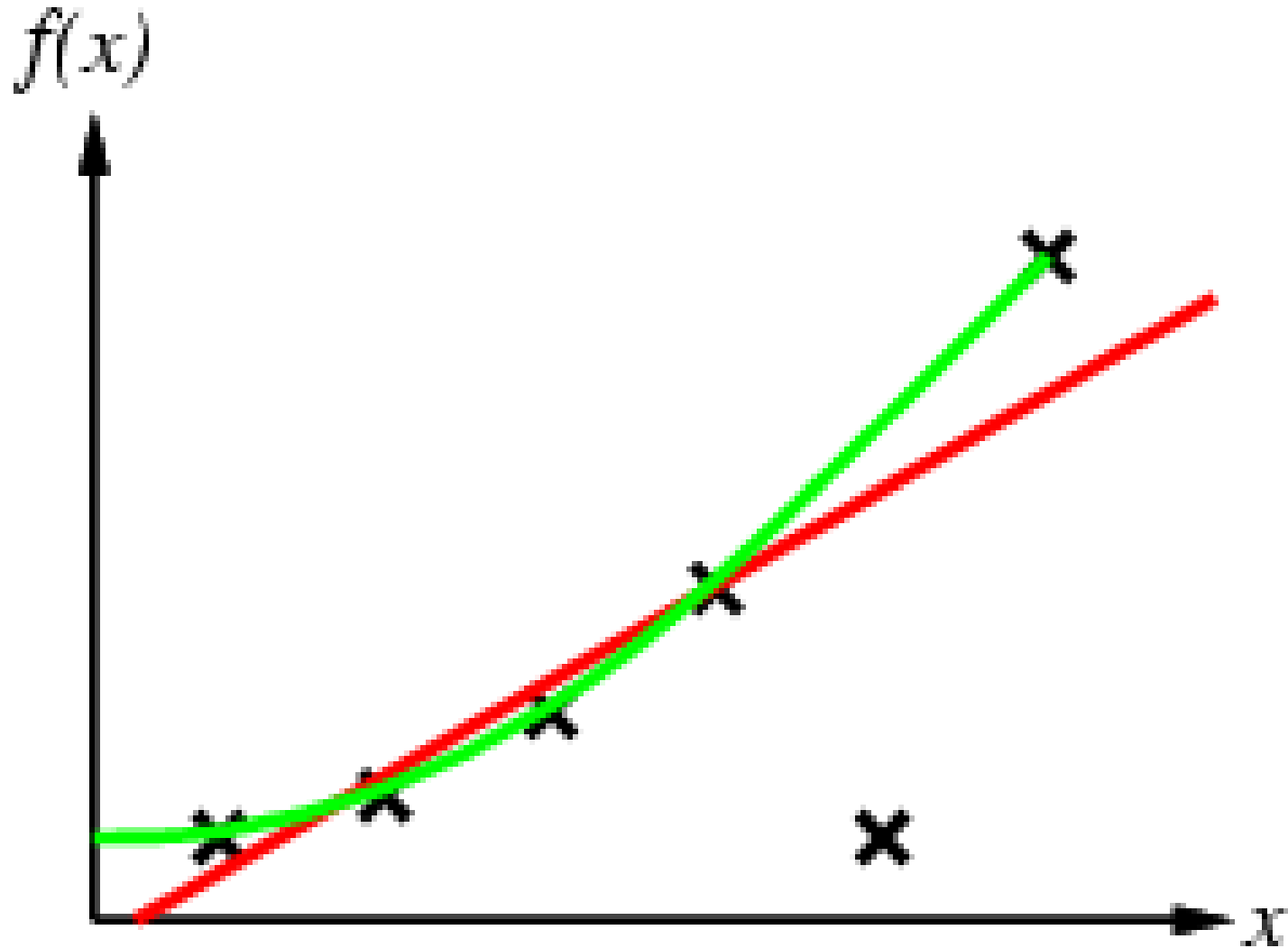
# Индуктивно учење (3)



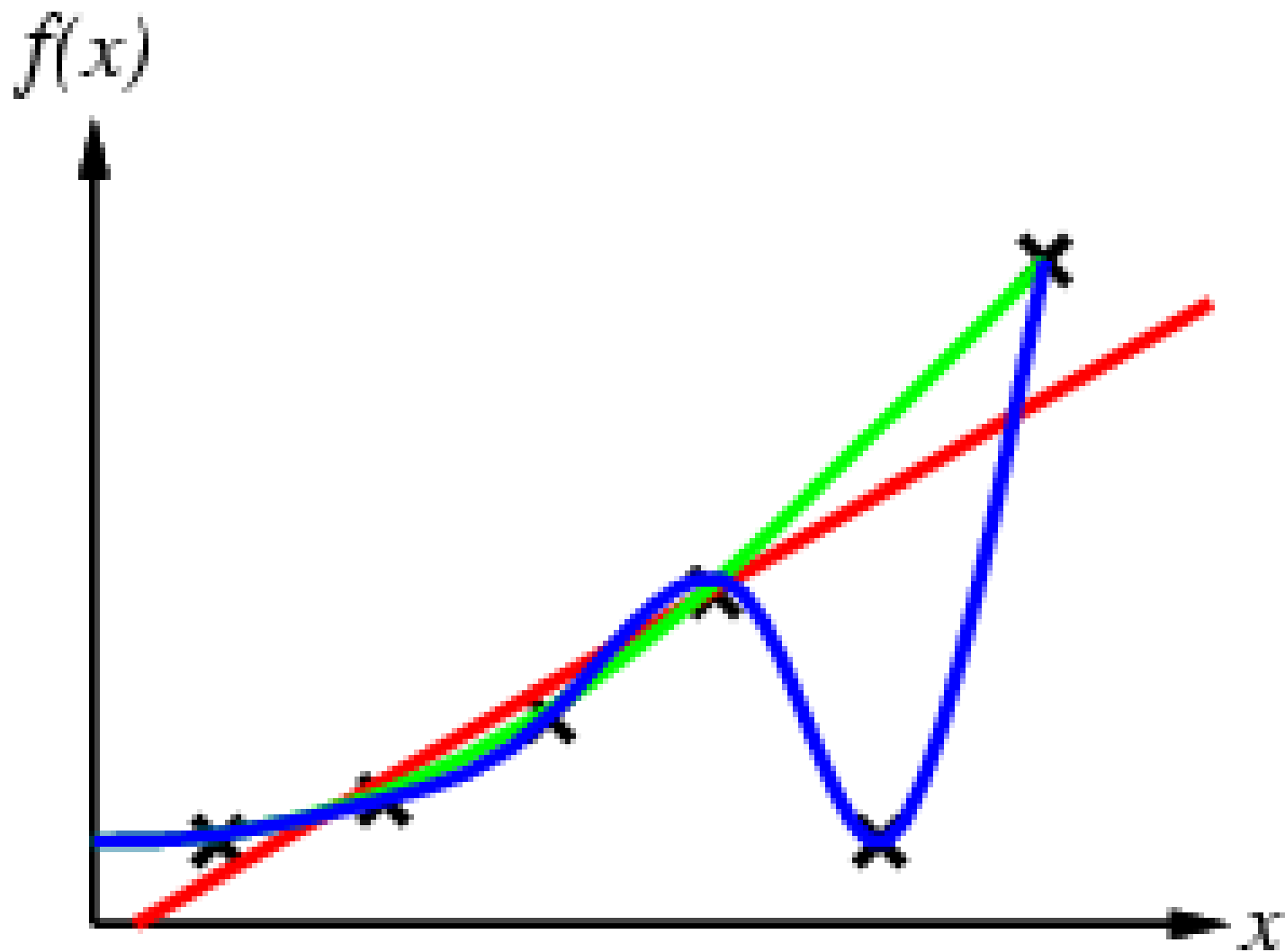
# Индуктивно учење (4)



# Индуктивно учење (5)



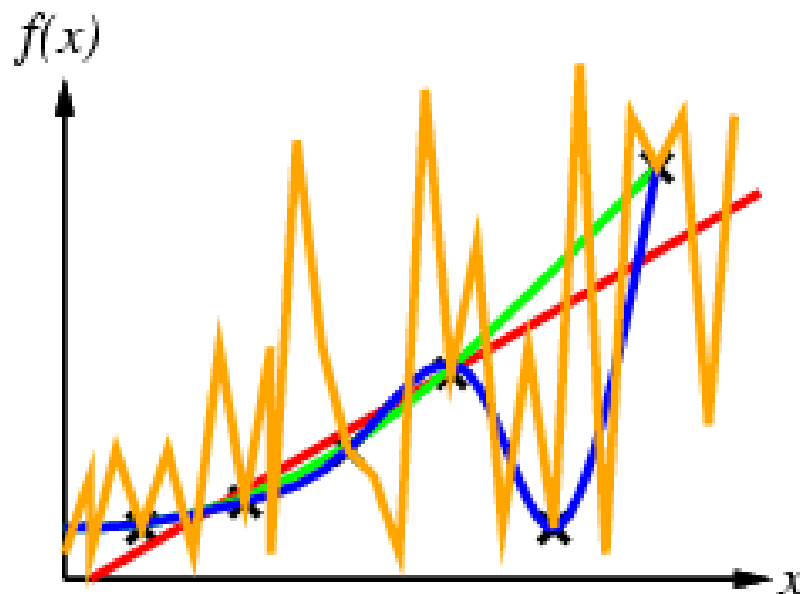
# Индуктивно учење (6)





# Индуктивно учење (7)

- bias – наклон. Сите алгоритми за учење изразуваат одреден вид на наклонетост кон една или друга хипотеза од множеството можни хипотези.



ричот на Окам (Ockham's razor): Преферирање на наједноставната хипотеза конзистентна со податоците.

# Инкрементално учење

- Наместо примена на алгоритмот за учење над целото множество на примероци за обука одеднаш, учење еден по еден примерок.
- Со презентацијата на секој нов примерок, агентот се обидува да ги измени постоечките хипотези (да бидат конзистентни и со новиот примерок и со сите поранешни). Агентот може да добива и одредени информации во врска со квалитетот на акциите кои ги презема.
- Учењето стихотворби е комбинација на инкрементално и целосно учење
- рва за одлучување (Decision Trees)
- Невронски мрежи (Neural Networks)
- SVM (Support Vector Machines) (машини со носечки / потпорни / поткрепувачки / поддржувачки вектори)

# AQ алгоритам (1)

- предложен од Michalski (1969).
- Влез: табела од примери кај кои се зададени вредностите на атрибутите.
- Излез: опис на концептот обично зададен со логички израз.

Табела Т. 6.1 Почетна табела за алгоритмот AQ				
пример	атриб1	атриб2	атриб3	припадност
E1	x	r	m	+ K
E2	y	r	n	+ K
E3	y	s	n	+ K
E4	x	s	m	- K
E5	z	t	n	- K
E6	z	r	n	- K

# AQ алгоритам (2)

1. аденото множество примери подели го на множество PE позитивни и NE негативни примери.
2. Одбери случајно пример од PE. Нека е тоа пример почетник, наречен семе.
3. Најди најгенерални описи (кои покриваат најмногу примери) на семето, при што границата е дефинирана со NE. Множеството на тие описи се нарекува везда.
4. Одбери некој опис од множеството описи, според некој критериум.
5. Ако така добиениот опис заедно со претходно одбраните описи ги покрива сите позитивни примери, запри; инаку најди друго семе и оди на 3.

# AQ алгоритам (3)

Табела Т. 6.1 Почетна табела за алгоритмот AQ

пример	атриб1	атриб2	атриб3	припадност
E1	x	r	m	+ K
E2	y	r	n	+ K
E3	y	s	n	+ K
E4	x	s	m	- K
E5	z	t	n	- K
E6	z	r	n	- K

итерација	чекор	акција
1	2	нека семе биде E2
1	3	најгенерален опис на E2 е (atrib1=y)
1	4	нема потреба
1	5	атриб1 не го покрива примерот E1. Нека е ново семе E1
2	3	најгенерален опис на E1 даваат описите (atrib1=x)&(atrib2=r) и (atrib2=r)&(atrib3=m)
2	4	преферираме (atrib1=x)&(atrib2=r)
2	5	добиваме опис (atrib1=y) V [(atrib1=x)&(atrib2=r)] Тој опис ги покрива сите позитивни примери Стоп.

Според примерот, концептот K ќе биде репрезентиран со изразот  $K = (atrib1=y) \vee [(atrib1=x) \wedge (atrib2=r)]$

# рва за одлучување

- Влез – објект или ситуација опишана со множество особини (обележја)
- Излез – А/НЕ одлука
- Секој внатрешен јазел претставува испитување на вредноста на една од особините (и разгранување во зависност од вредноста)
- Секој лист претставува булова вредност која треба да се врати како резултат ако се стигне до него

# Пример Russell & Norvig (1)

Problem: decide whether to wait for a table at a restaurant, based on the following attributes:

1. Alternate: is there an alternative restaurant nearby?
2. Bar: is there a comfortable bar area to wait in?
3. Fri/Sat: is today Friday or Saturday?
4. Hungry: are we hungry?
5. Patrons: number of people in the restaurant (None, Some, Full)
6. Price: price range (\$, \$\$, \$\$\$)
7. Raining: is it raining outside?
8. Reservation: have we made a reservation?
9. Type: kind of restaurant (French, Italian, Thai, Burger)
10. WaitEstimate: estimated waiting time (0-10, 10-30, 30-60, >60)

# Пример Russell & Norvig (2)

- Проблем: одлука дали да се чека за маса во некој ресторан, врз основа на следните атрибути:
  1. Алтернатива: дали има алтернативен ресторан во близина?
  2. бар: дали има удобен бар за чекање?
  3. Пет/Саб: дали денот е петок или сабота?
  4. Гладни: дали сме гладни?
  5. Стални гости: бројот на познати стални гости во ресторанот (ниеден, неколку, сите)
  6. Цена: опсег на цени (\$, \$\$, \$\$\$)
  7. Врне: дали надвор врне?
  8. Резервација: дали сме направиле резервација?
  9. Тип: тип на ресторанот (француски, италијански, тајландски, сендвичара)
  10. ПроценкаЧекање: проценето време на чекање (0-10, 10-30, 30-60, >60)



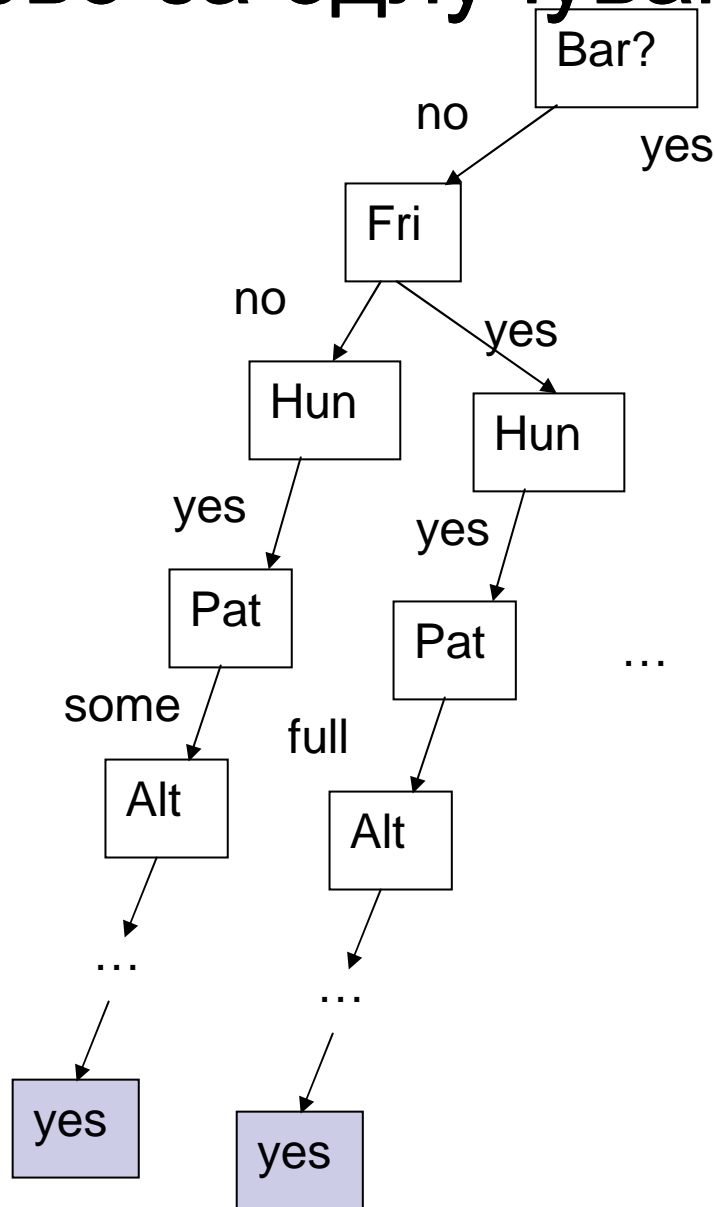
# Пример Russell & Norvig (3)

- Претстава на примероците преку вредности на атрибутите кои може да се дискретни, континуирани или улови
- Класификација на примероците на позитивни и негативни

Example	Attributes										Target
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	<i>Wait</i>
$X_1$	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
$X_2$	T	F	F	T	Full	\$	F	F	Thai	30–60	F
$X_3$	F	T	F	F	Some	\$	F	F	Burger	0–10	T
$X_4$	T	F	T	T	Full	\$	F	F	Thai	10–30	T
$X_5$	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
$X_6$	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
$X_7$	F	T	F	F	None	\$	T	F	Burger	0–10	F
$X_8$	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
$X_9$	F	T	T	F	Full	\$	T	F	Burger	>60	F
$X_{10}$	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
$X_{11}$	F	F	F	F	None	\$	F	F	Thai	0–10	F
$X_{12}$	T	T	T	T	Full	\$	F	F	Burger	30–60	T

# Пример на некое случајно избрано

## дрво за одлучување



Проблем: дрвото може да е многу големо, т.е. неоптимално

Колку различни дрва на одлука може да се креираат со  $n$  улови атрибути?

$n \Rightarrow$  број на атрибутите

$2^n \Rightarrow$  комбинации влезови

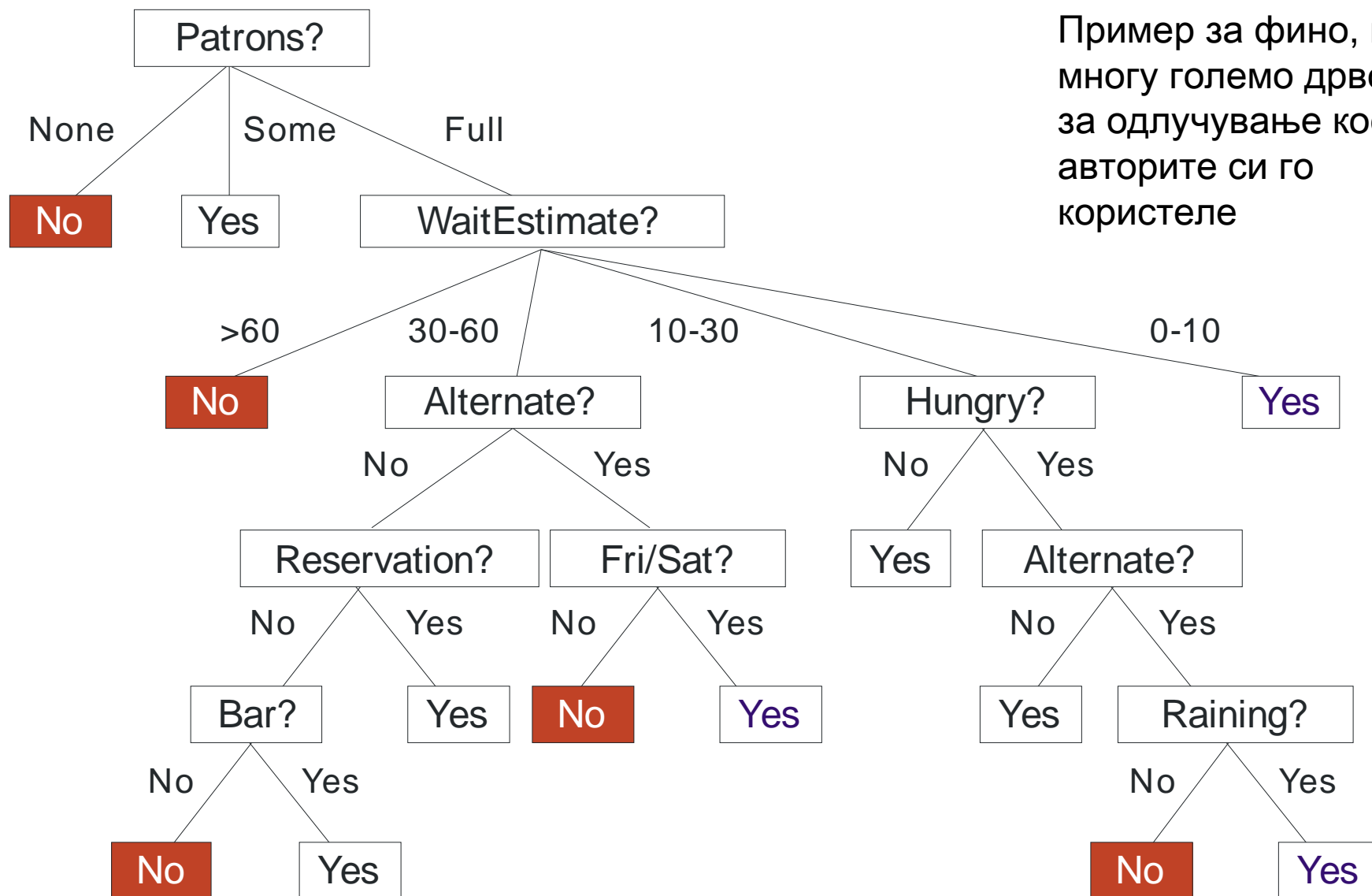
$2$  на степен  $2n \Rightarrow$  број на различни функции

(пр. со 6 улови атрибути постојат

18,446,744,073,709,551,616

дрва  $\approx 2 \times 10^{19}$

# Пример Russell & Norvig (4)



# Алгоритам за учење на дрвото за одлучување \*

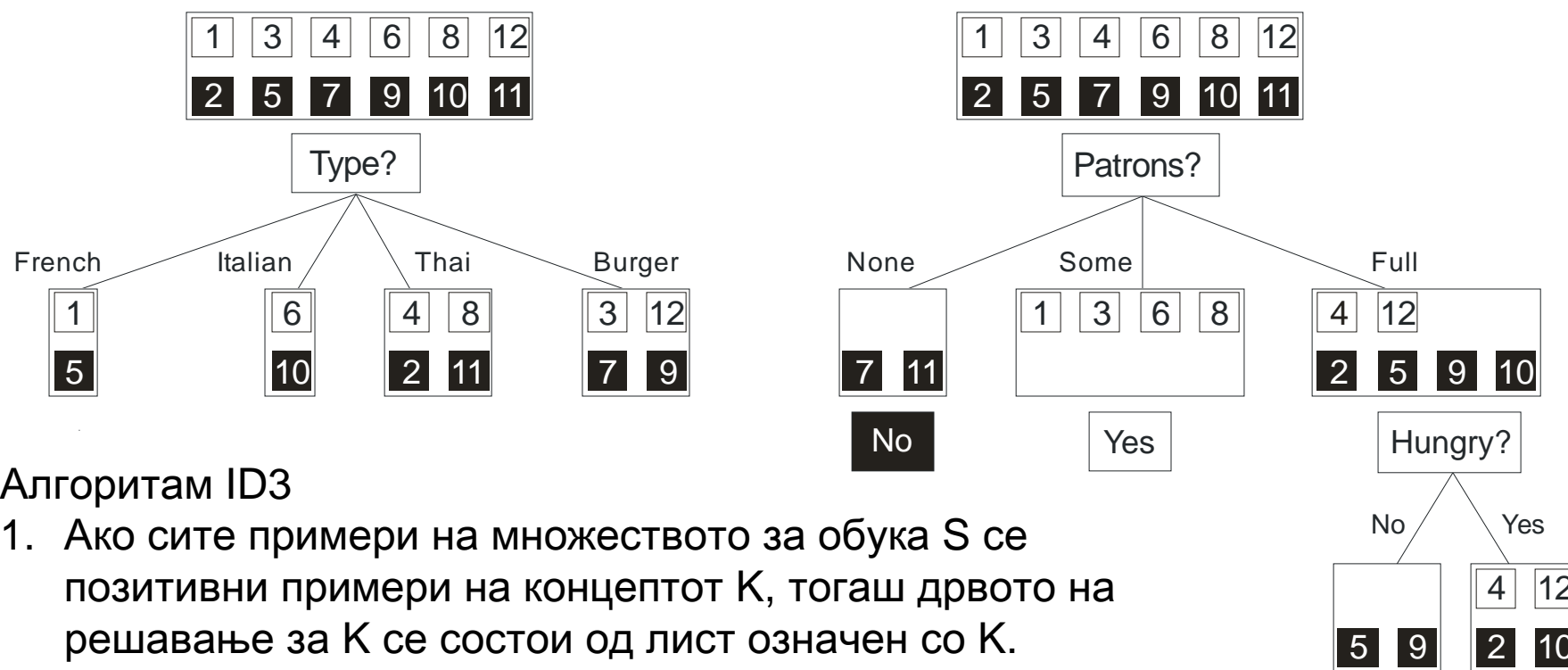
- Цел: да се пронајде мало дрво конзистентно со примероците за обука
- Идеја: рекурзивно избирање на „најзначајниот“ атрибут како корен на (под)дрво

```
function DTL(examples, attributes, default) returns a decision tree
  if examples is empty then return default
  else if all examples have the same classification then
    return the classification
  else if attributes is empty then return MODE(examples)
  else
    best  $\leftarrow$  CHOOSE-ATTRIBUTE(attributes, examples)
    tree  $\leftarrow$  a new decision tree with root test best
    for each value  $v_i$  of best do
      examplesi  $\leftarrow$  {elements of examples with best =  $v_i$ }
      subtree  $\leftarrow$  DTL(examplesi, attributes — best, MODE(examples))
      add a branch to tree with label  $v_i$  and subtree subtree
    return tree
```

# Учење на дрвото за одлучување

- Се бира “најдобриот” атрибут, т.е. оној што најмногу придонесува за лесната класификација на примероците.
- Ако се пола примероци негативни, а пола позитивни, тој атрибут не ни кажува многу, па не е добро него да го избереме на почетокот.
- Ако има само неколку позитивни или само неколку негативни примероци според некој атрибут, најдобро е таков некаков атрибут да го избереме погоре во дрвото и според него најпрво да се поделат останатите примероци.
- Ако остатокот се сите позитивни (или сите негативни) примероци тогаш сме дошле до крај (до лист) т.е. може да се одговори со А/НЕ.

# ID3 алгоритам



## Алгоритам ID3

1. Ако сите примери на множеството за обука  $S$  се позитивни примери на концептот  $K$ , тогаш дрвото на решавање за  $K$  се состои од лист означен со  $K$ .
2. Инаку, одбери тест кој ќе го подели множеството за обука  $S$  во две или повеќе подмножества  $S_i$ , зависно од излезот на тестот (на пр. вредностите на атрибутите). Тестот станува корен на дрвото и за секој излез на тестот изгради поддрво, повикувајќи ја оваа процедура рекурзивно за секое  $S_i$ .

# Теорија на информации

- а се имплементира Choose-Attribute во DTL алгоритмот
- Количество информација (Entropy):

$$I(P(v_1), \dots, P(v_n)) = \sum_{i=1}^n -P(v_i) \log_2 P(v_i)$$

- За множество за обука со  $p$  позитивни и  $n$  негативни примери:

$$I\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} \log_2 \frac{p}{p+n} - \frac{n}{p+n} \log_2 \frac{n}{p+n}$$

# Информациска добивка

- Избраниот атрибут  $A$  го дели множеството за обука  $E$  на подмножества  $E_1, \dots, E_v$  според вредностите на атрибутот  $A$ , каде  $A$  прима  $v$  различни вредности.

$$remainder(A) = \sum_{i=1}^v \frac{p_i + n_i}{p + n} I\left(\frac{p_i}{p_i + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- Информациска добивка (Information Gain - IG) - намалувањето на ентропијата по изборот на атрибутот  $A$ :

$$IG(A) = I\left(\frac{p}{p + n}, \frac{n}{p + n}\right) - remainder(A)$$

- Избери го атрибутот со најголем IG



# Информациска добивка\*

За множеството за обука,  $p = n = 6$ ,  $I(6/12, 6/12) = 1$  bit

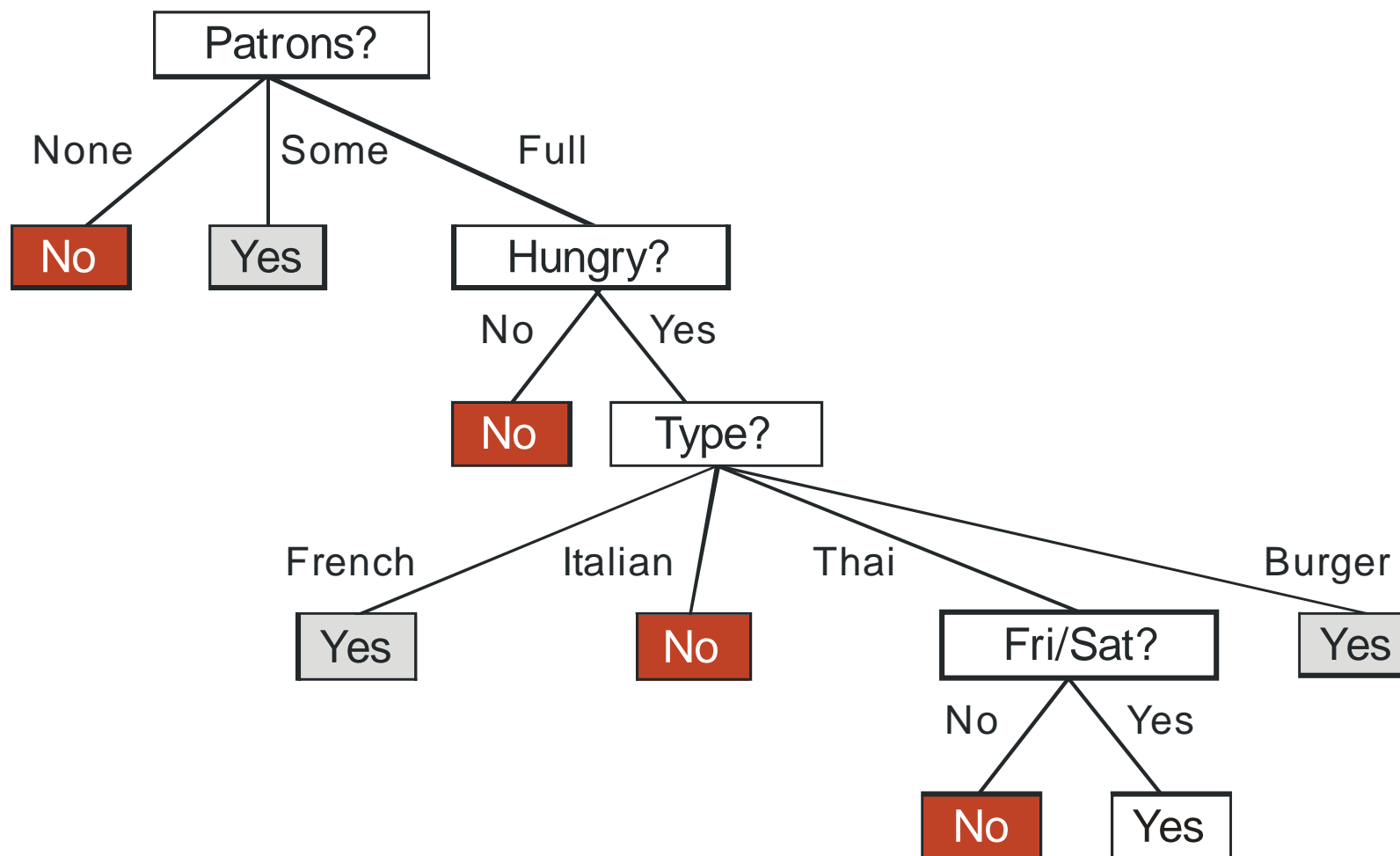
а ги разгледаме атрибутите *Patrons* и *Type* (и останатите):

$$IG(Patrons) = 1 - \left[ \frac{2}{12} I(0,1)^{\text{none}} + \frac{4}{12} I(1,0)^{\text{some}} + \frac{6}{12} I\left(\frac{2}{6}, \frac{4}{6}\right) \right] = .0541 \text{ bits}$$

$$IG(Type) = 1 - \left[ \frac{2}{12} I\left(\frac{1}{2}, \frac{1}{2}\right)^{\text{French}} + \frac{2}{12} I\left(\frac{1}{2}, \frac{1}{2}\right)^{\text{Italian}} + \frac{4}{12} I\left(\frac{2}{4}, \frac{2}{4}\right)^{\text{Thai}} + \frac{4}{12} I\left(\frac{2}{4}, \frac{2}{4}\right)^{\text{Burger}} \right] = 0 \text{ bits}$$

*Patrons* има највисок IG од сите атрибути и е избран од DTL алгоритмот како корен

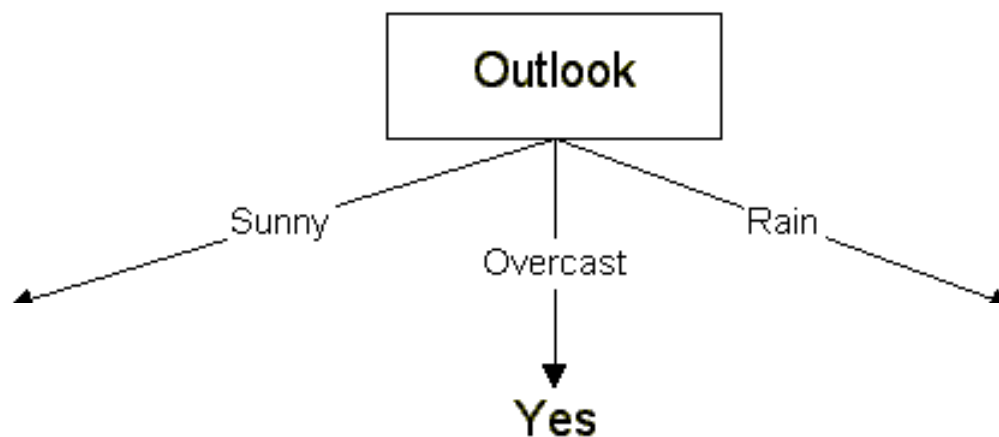
# обиено дрво за одлучување



# Пример 2: али да се игра со топка? (1)

Day	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

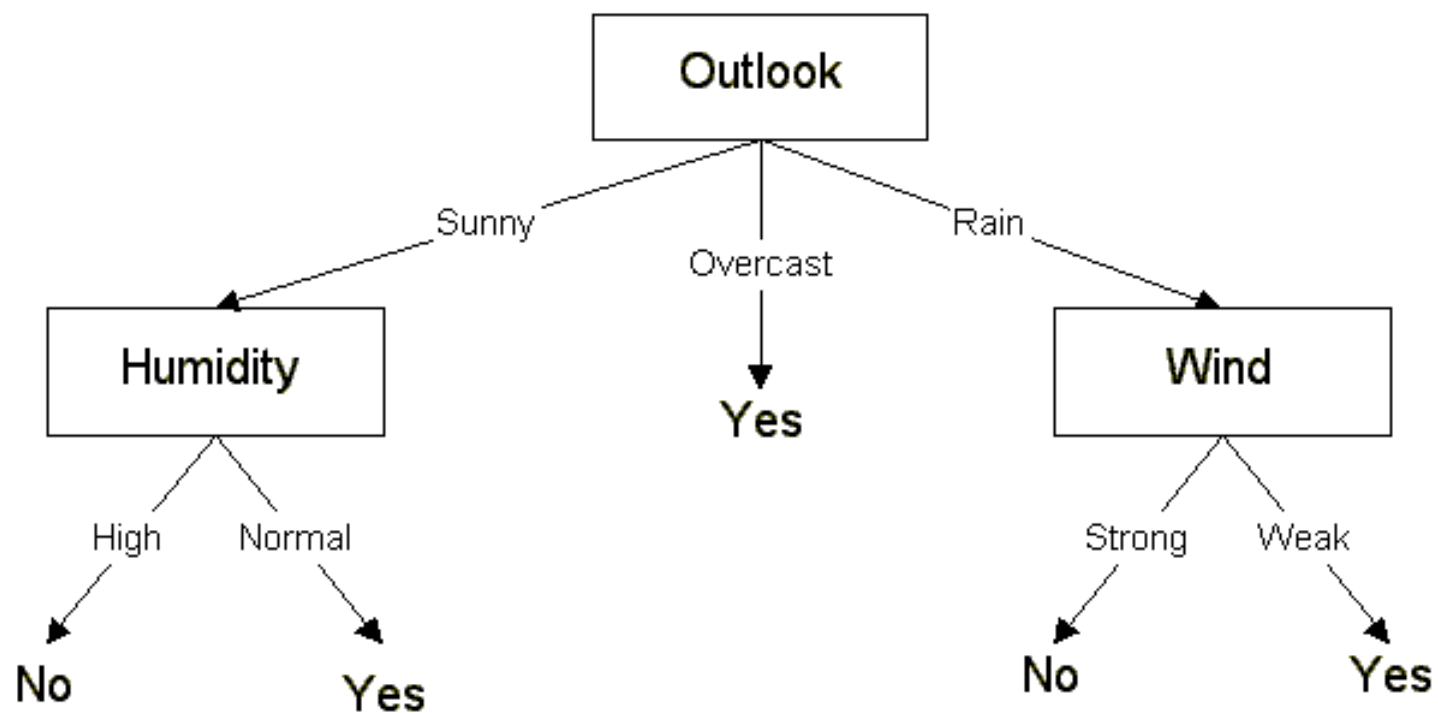
# Научено дрво (2)



# Пример 2: али да се игра со топка? (3)

Outlook	Day	Temperature	Humidity	Wind	Play ball
Sunny	D1	Hot	High	Weak	No
Sunny	D2	Hot	High	Strong	No
Sunny	D8	Mild	High	Weak	No
Sunny	D9	Cool	Normal	Weak	Yes
Sunny	D11	Mild	Normal	Strong	Yes
Rain	D4	Mild	High	Weak	Yes
Rain	D5	Cool	Normal	Weak	Yes
Rain	D6	Cool	Normal	Strong	No
Rain	D10	Mild	Normal	Weak	Yes
Rain	D14	Mild	High	Strong	No
Overcast	D3	Hot	High	Weak	Yes
Overcast	D7	Cool	Normal	Strong	Yes
Overcast	D12	Mild	High	Strong	Yes
Overcast	D13	Hot	Normal	Weak	Yes

# Научено дрво (4)

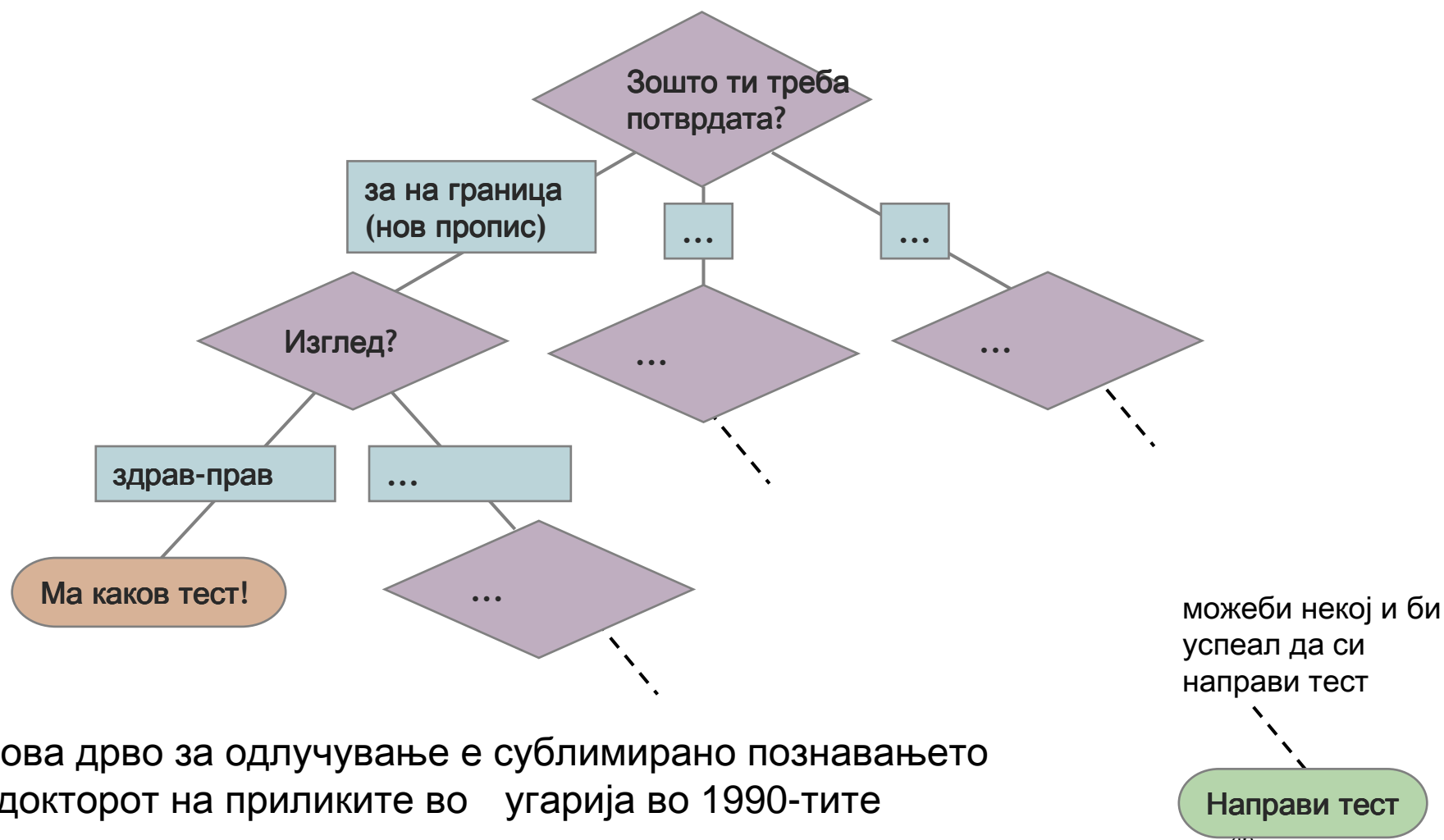




# Зошто дрва за одлучување? (1)

- Целта е со што помалку прашања да се одлучи за нешто
- Кога треба да одлучиме за некоја комбинација која до сега не се појавила

# Како во угарија прават тест за СИ А 😊



Во ова дрво за одлучување е сублимирано познавањето на докторот на приликите во угарија во 1990-тите



# Зошто дрва за одлучување? (2)

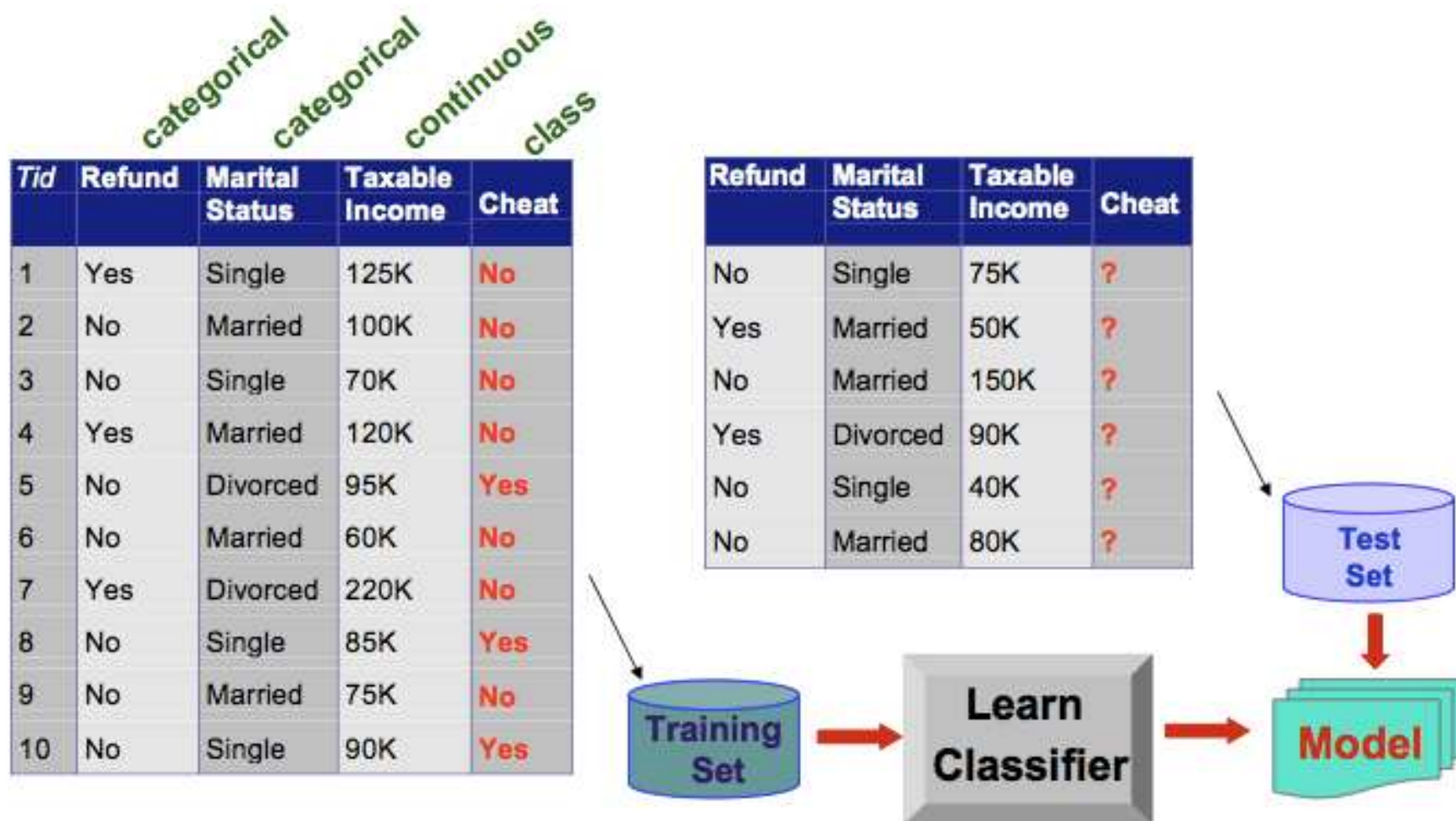
- Лесно интуитивно се толкуваат
- C4.5 (j48) се многу популарни
- Реален пример (од Russell & Norvig)
  - Системот на британскиот Петрол (BP) за одвојување на гас од нафта, што имал рачно направен потсистем со 2500 правила, бил заменет со дрва за одлучување. Системот направен со C4.5 давал подобри перформанси од човечките експерти и му заштедил на BP милиони. (1986)

# Проблеми кај дрвата за одлучување

- Ако за некоја комбинација нема примерок во базата, т.е. значи нема опсервација за таков пример, тогаш се враќа некоја предодредена (default) вредност.
- Ако нема веќе атрибути, но имаме сеуште позитивни и негативни примероци, настанува проблем. Тоа значи дека примероците имаат ист опис, но различна класификација. Тоа се случува кога недостасуваат некои од податоците или се погрешни. Може да се случи кога проблемот е недетерминистички или кога не е опишан целосно со дадените примероци. Тогаш може да се користи мнозинско гласање - majority vote.

# Трите основни множества кај учењето

- Учењето кое се базира на множества, ги користи овие три множества:
  - Множество за обука (training set)
  - Множество за исправност (validation set)
  - Множество за тестирање (testing set)
- Овие три множества не се неопходни во сите системи способни да учат
- Тие се карактеристични за индуктивното учење и за невронските мрежи



Класификација: множеството за обука се состои од група на записи кои содржат атрибути од кои едниот атрибут е класа. Задачата е да се најде моделот на класниот атрибут како функција од другите атрибути. Множеството за тестирање се користи за валидација на моделот.

# Множество за обука

- ефиниција: множество примери кои се користат за да се научи системот да генерира точен одговор.
- Постапка:
  - Од популацијата се одбира примерок кој е претставен во форма на влезен вектор за кој се генерира вектор одговор.
  - Притоа, целта е одговорот да се совпадне со излезот, т.е. да се минимизира средно-квадратното отстапување на излезот и одговорот, кое ја претставува грешката на обучувањето.
  - Обучувањето завршува кога грешката ќе биде помала од однапред дефинираната.

# Множество за исправност

- ефиниција: множество примери кои се користат за да се нагодат параметрите дефинирани при обучувањето.
- Постапка:
  - ☐ Множеството за исправност е множество независно од множеството за обука.
  - ☐ Со него се проверува исправноста на предвидените релации воспоставени при обучувањето.
  - ☐ Грешката при генерирањето на одговорот треба да биде помала од вредност што се дефинира однапред.
  - ☐ околку овој критериум не е исполнет, тогаш треба да се нагодат параметрите на обучувањето и да се повтори обучувањето со претходно одбраното множество за обука.

# Множество за тестирање

- ефиниција: множество примери со кои се мери однесувањето, односно способноста за генерализација на обучениот систем.
- Постапка:
  - По воспоставувањето на релациите кои се одобрени со тестот за исправност, се одбираат примери различни од двете претходни.
  - За секој влезен пример се генерира одговорот и се споредува со очекуваниот излез.
  - Со тоа се проценува успешноста на способноста за генерализација

# Што се прави кога нема доволно податоци

- Во отсуство на доволно податоци, множеството за обучување се разделува на две подмножества, едно кое ќе се користи за обука, а второ за исправност.
- Ова разделување се вика вкрстена исправност (cross-validation).



# Техники за вкрстената исправност

- Случајно одбирање (random sampling)
  - основното множество случајно се дели на два дисјунктни дела со различна големина, од кои едното е за обука, а другото за исправност
- К-кратна поделба (K-fold cross validation)
  - основното множество случајно се дели на  $K$  дисјунктни делови со сличен обем, од кои  $K-1$  се за обука, а последното за исправност
  - притоа, секое од  $K$ -те множества еднаш се зема за тестирање, што значи дека постапката се повторува  $K$  пати
- Поделба со оставање на еден (leave-one-out cross validation)
  - ако множеството е многу мало за да се дели на  $K$  делови, во тој случај,  $n-1$  примери од вкупно  $n$ -те достапни се користат за обука, а само еден за тестирање и постапката се повторува  $n$  пати

# Пример за македонскиот јазик

- Цел: учење на правила според кои се збороформираат именките (т.е. како од именката дете да се добијат збороформите: детето, детево, детено, деца, децата, децава, децана)
- Множество за обука и оценка на исправноста: сите именки од една книга.
- Разделување на овие множества: 10fold crossvalidation, т.е. случајно делење на сите именки на 10 групи, од кои едната се користи за обука, а останатите за исправност.
- Тестирање: именките од речникот кои ги нема во книгата.

# Користена литература

- Artificial Intelligence, A Modern Approach  
2nd edition, Russel and Norvig
- ожиновски С., Вештачката  
интелигенција, Гоцмар, Скопје, 1994



# Прашања?

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Невронски мрежи

# Невронски мрежи

- Два погледа на невронските мрежи:
  - од пресметковен (компјутерски) аспект – вештачки невронски мрежи – метод за претстава на функции со помош на едноставни аритметички пресметковни елементи (и нивна обука од примероци)
  - од биолошки аспект - математички модел на работата на мозокот
- Неврони – клетки кои ја изведуваат обработката на информациите во мозокот
- Мрежа од меѓусебно поврзани неврони – невронска мрежа

# Што се тоа „вештачки невронски мрежи“ (Artificial Neural Networks)?

- Парадигма за обработка на информации инспирирана од начинот на кој природните мозоци ја обработуваат информацијата, составени од густо меѓуповрзани паралелни структури
- Вид на повеќепроцесорски системи со:
  - Едноставни процесорски елементи
  - Висок степен на меѓусебна поврзаност
  - Едноставни пораки со броеви
  - Прилагодлива интеракција помеѓу елементите

# Мозок

- Како работи мозокот (мистерија?)
- 335 пр.н.е. Аристотел - „Од сите животни, човекот има најголем мозок во однос на неговата големина“
- Неврон – фундаментална функционална единица на сите нервни ткива



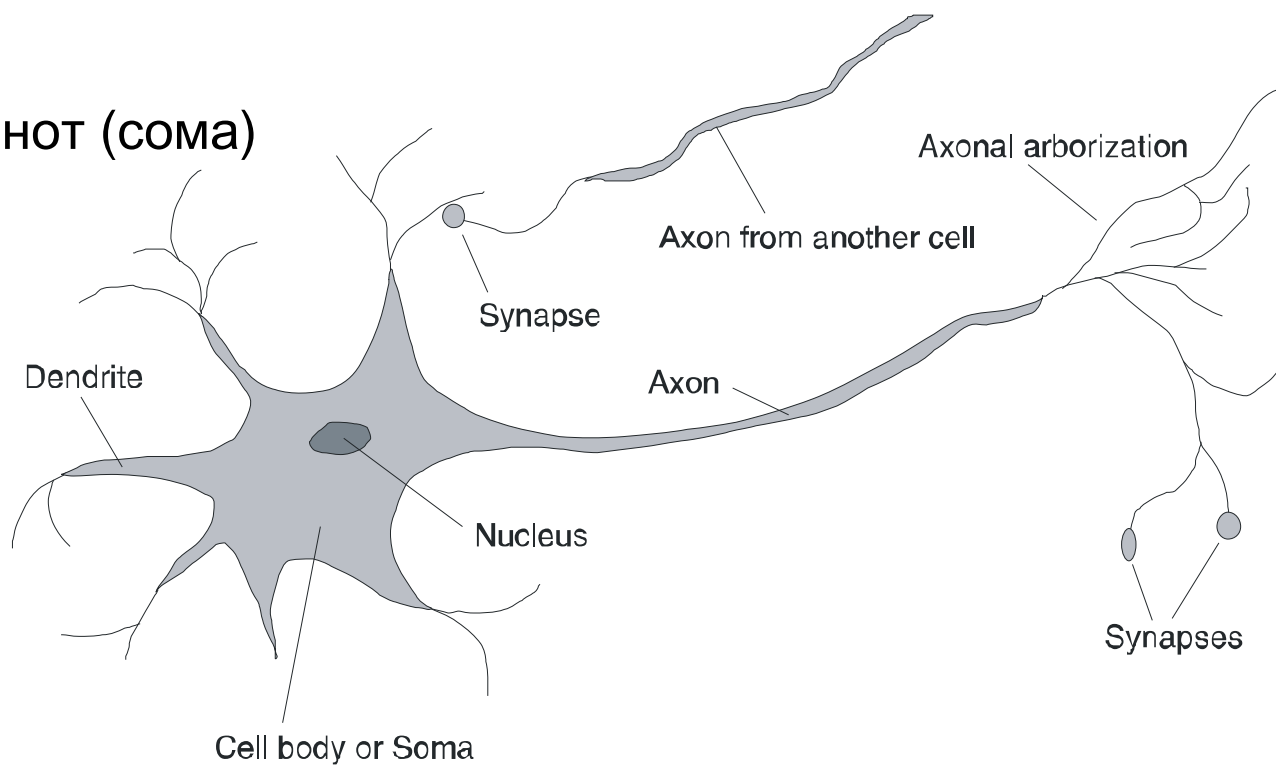
# Работа на неврнонот

- Сложен електрохемиски процес
- Акциски потенцијал – ниво на активација
- Синапси - ексцитациски (поттикнувачки) / инхибиторни (спречувачки)
- Пластичност

# Неврон

## ■ Структура:

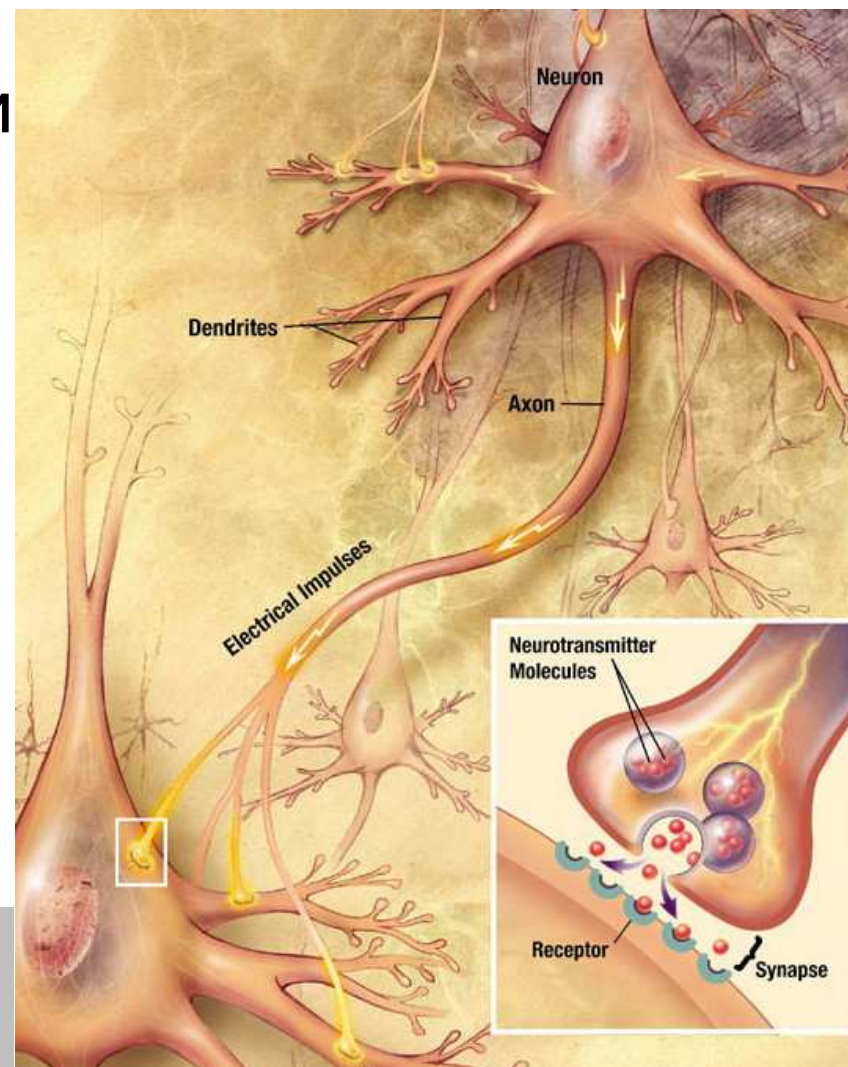
- ☐ Тело на невронот (сома)
- ☐ Аксон
- ☐ Дендрити
- ☐ Синапси



# Работа на невронот - синапси

- Сумарен постсинаптички потенцијал (СПП)
- Синаптичко влијание
  - ☐ екситациско
  - ☐ инхибиторно
- Пластичност

Синапсите се основата за меморирањето и учењето



# Споредба

- Кај мозокот сите неврони работат симултано

	Computer	Human brain
Computational units	<sup>62</sup> <del>1</del> CPU, <sup>10<sup>9</sup></sup> <del>10<sup>8</sup></del> gates	10 <sup>11</sup> neurons
Storage Units	10 <sup>10</sup> bits RAM 10 <sup>11</sup> bits disk	10 <sup>11</sup> neurons 10 <sup>14</sup> synapses
Cycle time	10 <sup>-9</sup> sec	10 <sup>-3</sup> sec
Bandwidth	10 <sup>10</sup> bits/sec	10 <sup>14</sup> bits/sec
Memory update/sec	10 <sup>9</sup>	10 <sup>14</sup>

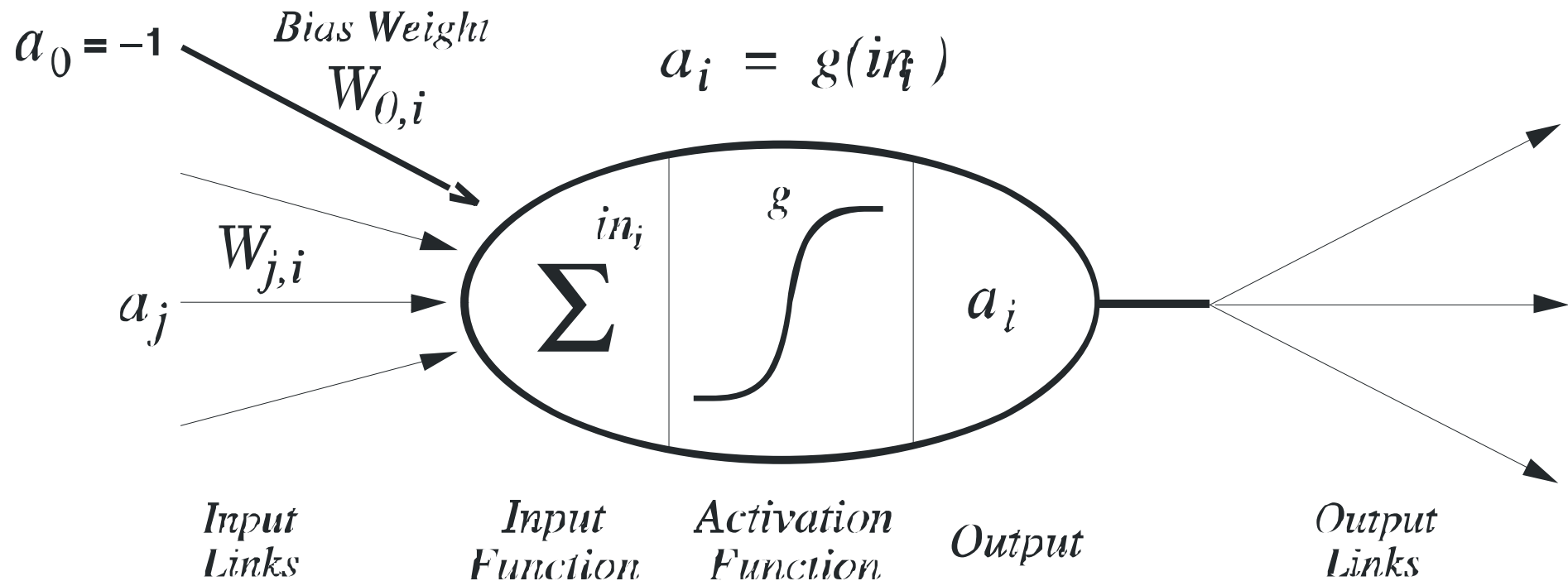
- брз
- точен
- (претежно)  
секвенцијален
- (брзо и точно)  
пресметување сложени  
математички изрази

- бавен
- fault-tolerant
- паралелен
- брзо препознавање на  
комплексни слики (сцени)

# Невронски мрежи

- Неврони (јазли), врски, тежини, активациско ниво (праг)
- За да се направи невронска мрежа која ќе решава одредена задача потребно е:
  - ☐ да се одреди бројот на невроните
  - ☐ нивниот тип
  - ☐ како ќе бидат поврзани
  - ☐ да се иницијализираат тежините и со некој алгоритам да се обучи мрежата претставувајќи и познати примероци
  - ☐ да се одлучи како ќе биде претставен проблемот (влез / излез)

# Вештачки неврон

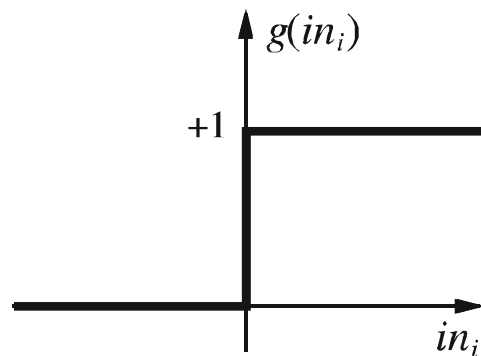


$$in_i = \sum_j W_{j,i} a_j = \mathbf{W}_i \cdot \mathbf{a}_i$$

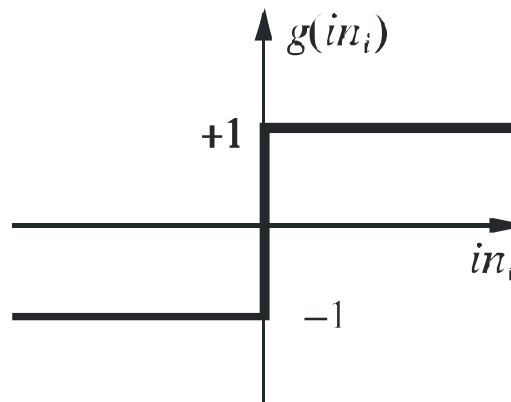
# Активациска функција

- Активација 
$$a_i \leftarrow g(in_i) = g\left(\sum_j W_{j,i} a_j\right)$$

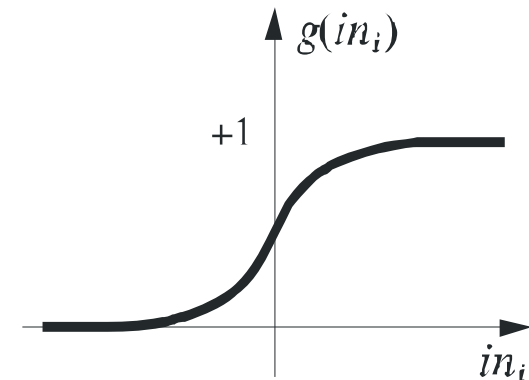
- Различни функции  $g$  – различни модели на неврони



$$step_t(x) = \begin{cases} 1, & \text{if } x \geq t \\ 0, & \text{if } x < t \end{cases}$$



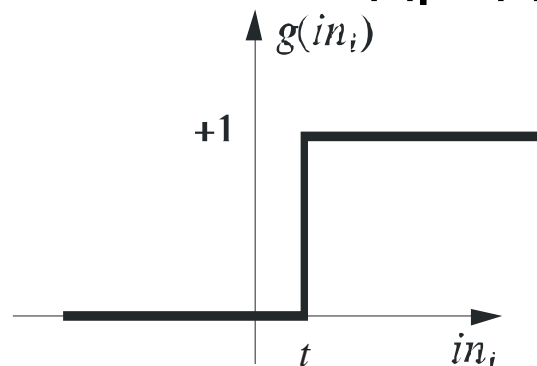
$$sign(x) = \begin{cases} +1, & \text{if } x > 0 \\ -1, & \text{if } x < 0 \end{cases}$$



$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

# Праг на активација

- Практично: Замена на прагот со дополнителен влез со одредена тежина



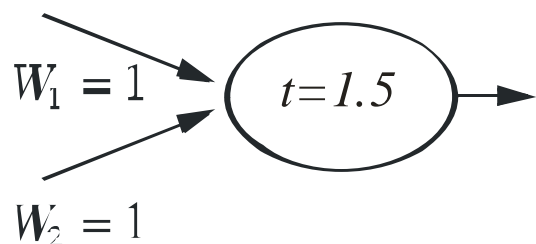
$$a_i = step_t \left( \sum_{j=1}^n W_{j,i} a_j \right) = step_0 \left( \sum_{j=0}^n W_{j,i} a_j \right)$$

каде  $W_{0,i} = t$  и  $a_0 = -1$

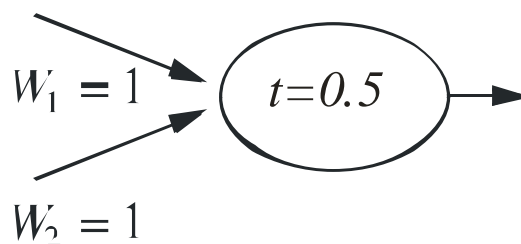


# Логички функции со невроните

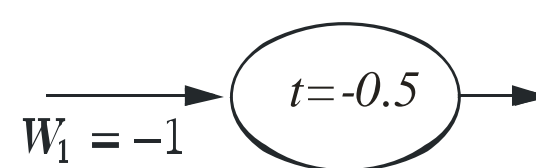
- Модели на неврони кои ги реализираат основните логички функции



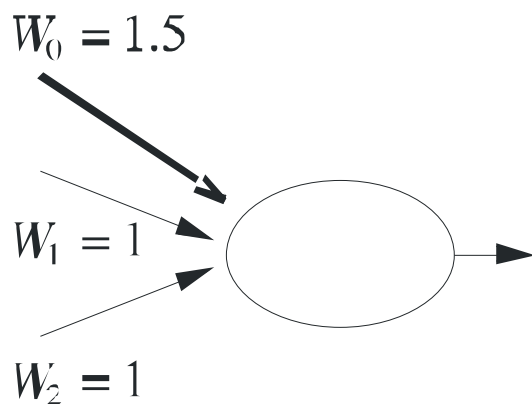
**AND**



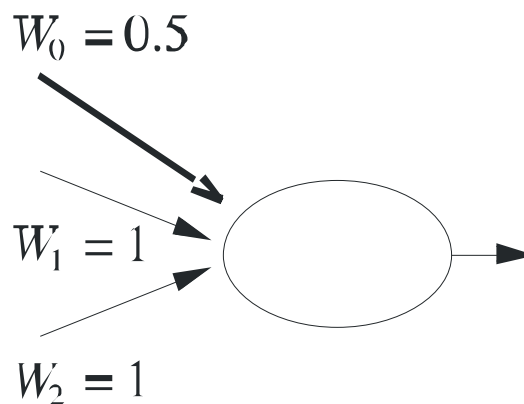
**OR**



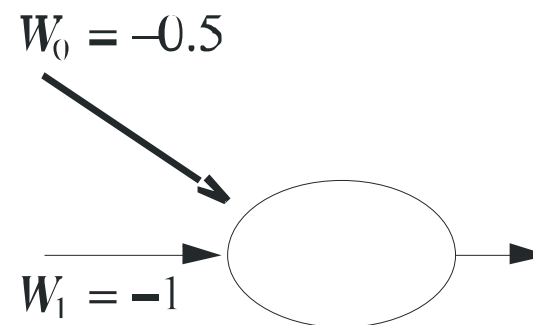
**NOT**



**AND**



**OR**



**NOT**

# Наједноставно учење кај НМ

## ■ Хебово учење

- Во 1949, Доналд Хеб напишал: „Кога аксонот на клетката А е доволно блиску до клетката Б за да може да ја ексцитира (возбуди) и повторливо и истрајно учествува во нејзиното палење (активирање), некои процеси на растење и метаболитски промени се случуваат во едната или во двете клетки, така што ефикасноста на А во возбудувањето на Б е зголемена“

# Хебово учење (Hebbian Learning)

- Психолозите и когнитивните научници ја генерализираа оваа идеја така што кога два перцепти повторливо се набљудуваат заедно, во меморијата се формира асоцијација помеѓу нив, така што едниот перцепт може да го повика (invoke) другиот.



# Формализација на Хебовото учење

- Изградба на асоцијации заради истовремено појавување (Association-Building by Co-occurrence)
- Формализирано како вид на ненадгледувано учење кај НМ
- $\Delta w_{ij} = \eta \cdot a_i \cdot a_j$ 
  - промена на тежината помеѓу невроните  $i$  и  $j$  со степен на учење (learning rate)  $\eta$

# Пример за Хебово учење (1)\*

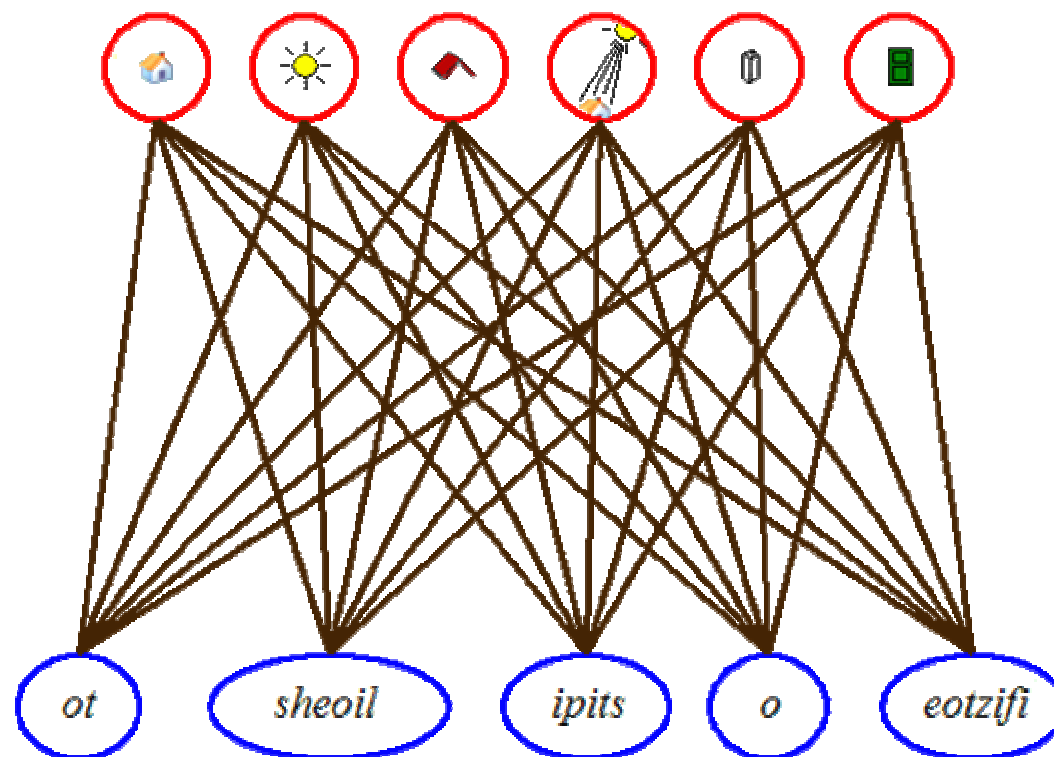
- Замислете дека треба да научите некој странски јазик (како што при средбата на Индијанците и Европејците се учеле странските јазици, на почетокот).
- Визуелниот и аудиторниот влез се даваат во парови на слика со реченица која веројатно ја објаснува таа слика



*o sheoil eotzifi ot ipits*

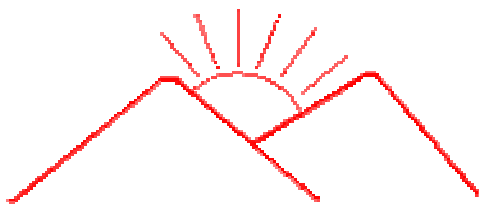
# Пример за Хебово учење (2)\*

- Формирање на асоцијации помеѓу секој визуелен перцепт и секој лингвистички симбол (збор)



# Пример за Хебово учење (3)\*

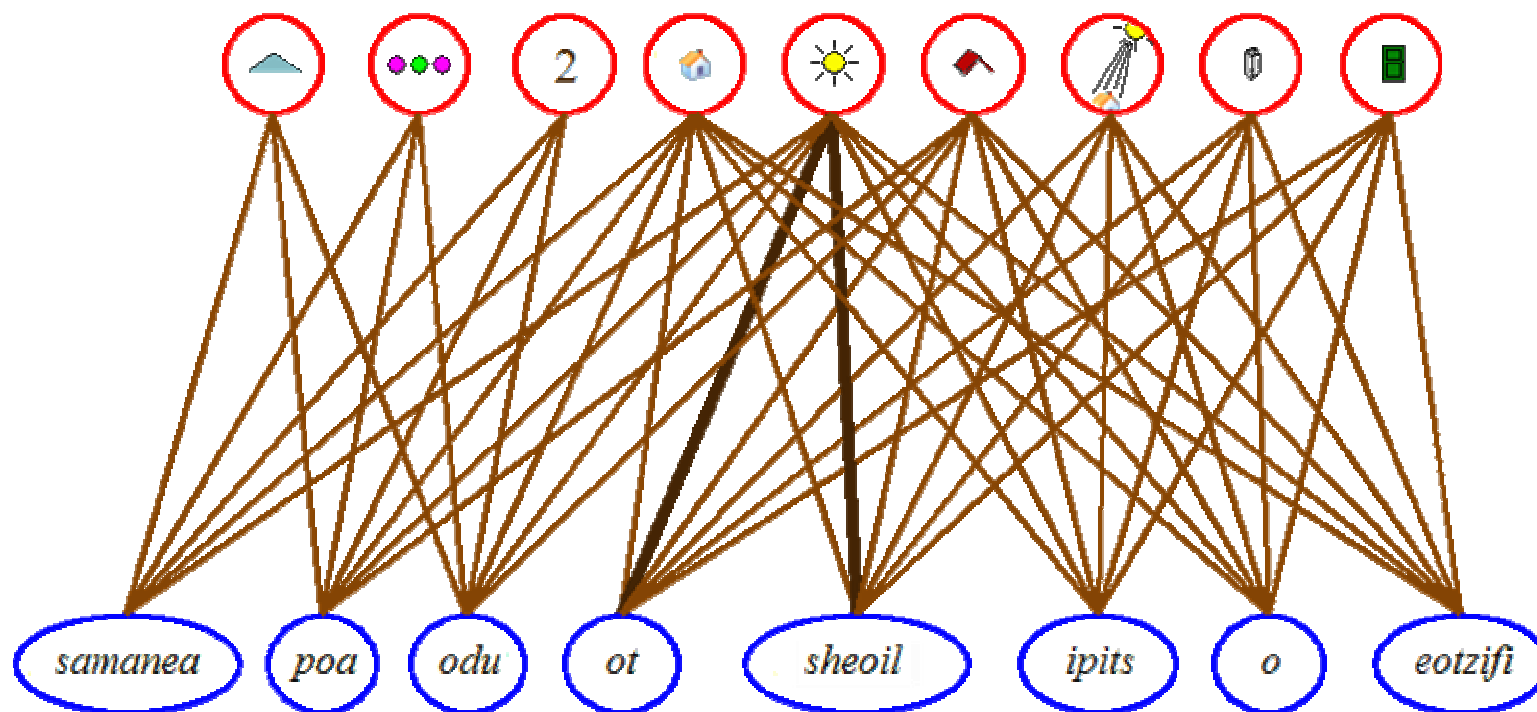
- Сега добивате нов пар на слика и реченица



*o sheoil ot eotzifi samanea poa odu onbau*

# Пример за Хебово учење (4)\*

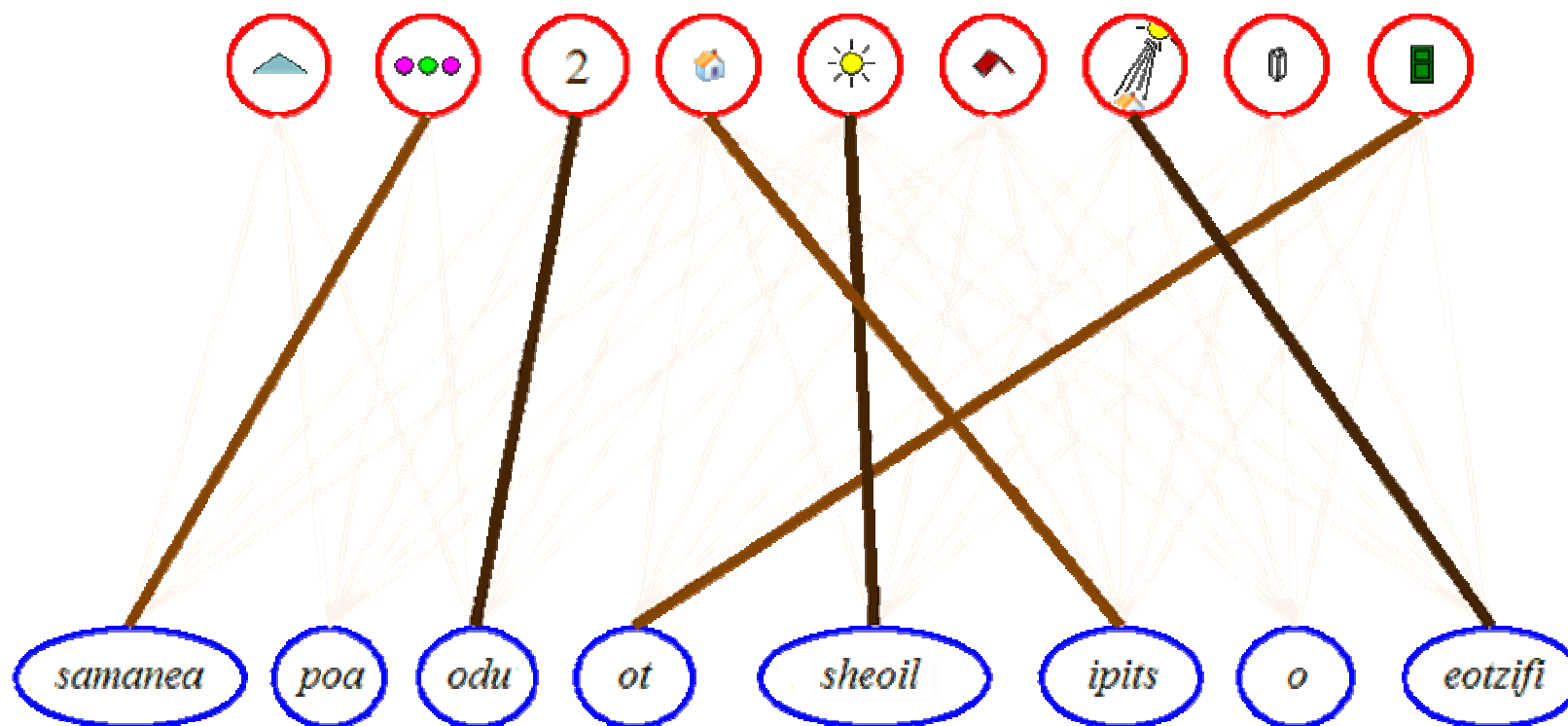
- Сега сликите и зборовите се додаваат како унија во соодветните множества и тие асоцијации каде повторно се појавија истите слики и зборови се засилуваат за разлика од другите кои се заслабнуваат



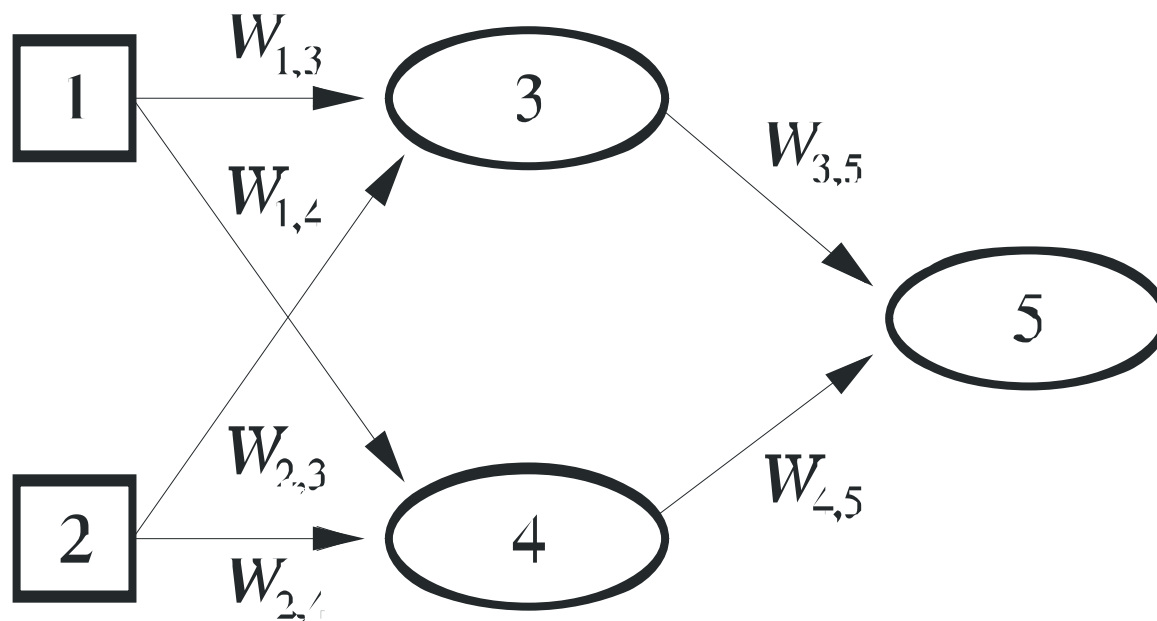


# Пример за Хебово учење (5)\*

- И така, со тек на време, со прикажување на многу парови на слики и реченици, ќе се издвојат (правилните) асоцијации помеѓу сликите и зборовите
- Колку ќе бидат правилни асоцијациите зависи од учителот



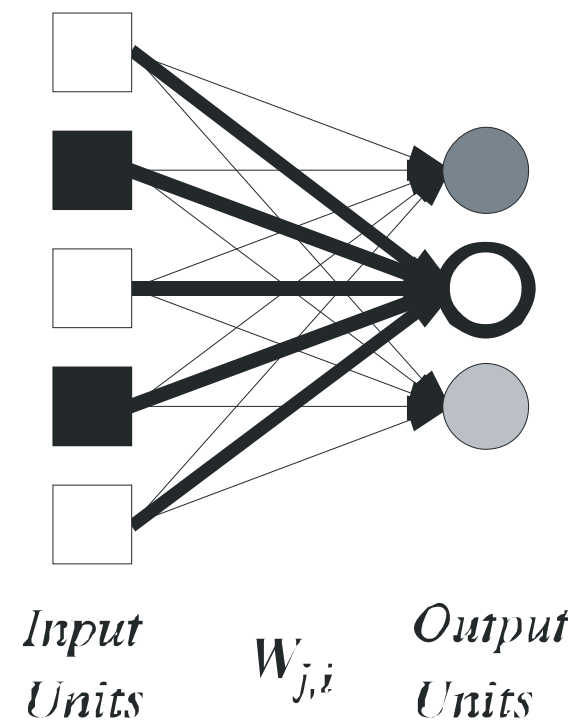
# Двослојна невронска мрежа



$$\begin{aligned} a_5 &= g(W_{3,5}a_3 + W_{4,5}a_4) \\ &= g(W_{3,5}g(W_{1,3}a_1 + W_{2,3}a_2) + W_{4,5}g(W_{1,4}a_1 + W_{2,4}a_2)) \end{aligned}$$

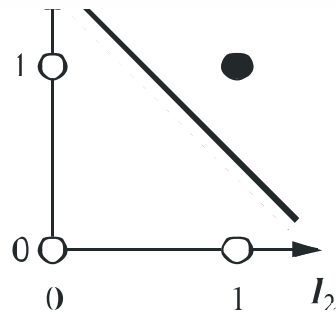
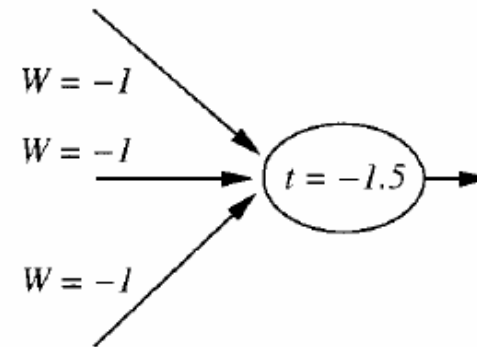
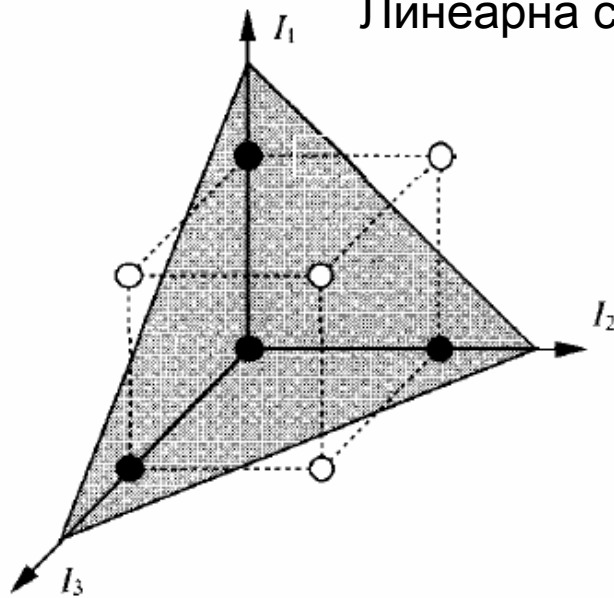
# Перцептрон

- Перцептрон = feed-forward невронска мрежа со едно ниво
- Што може да репрезентираат перцептроните?
  - majority може, XOR не може
- Перцептронот може да репрезентира само линеарно сепарабилни функции!

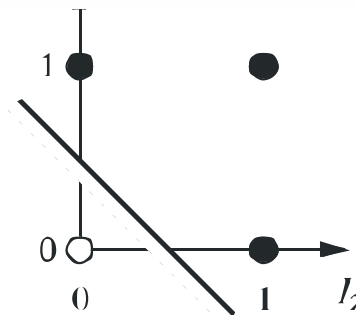


# Ограничувања на перцептронот

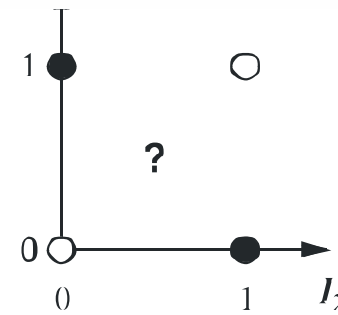
Линеарна сепарабилност (делливост) во хипер-рамнина



(a)  $I_1$  and  $I_2$



(b)  $I_1$  or  $I_2$



(c)  $I_1$  xor  $I_2$

# Обука на перцептронот

- Постои алгоритам кој може да научи било која линеарно сепарабилна функција ако му се дадат доволно примероци за обука
  1. Иницијализација на тежините на случајни вредности
  2. Промена на овие вредности со цел да се направат конзистентни со примерите – итеративен процес на мали промени на тежините со кои би се намалила разликата помеѓу добиениот и бараниот излез за даден примерок - конвергенција

# Правило за промена на тежините

- Епоха – промена на сите тежини за сите примероци
- Правило за промена на тежините

$$Err = T - O$$

- Секој влез придонесува  $W_j I_j$  на вкупниот излез

$$W_j \leftarrow W_j + \alpha \cdot I_j \cdot Err$$

- $\alpha$  - стапка на обука (learning rate)

# Generic NN learning – општо учење кај НМ\*

**function** NEURAL-NETWORK-LEARNING(*examples*) **returns** *network*

*network*  $\leftarrow$  a network with randomly assigned weights

**repeat**

**for each** *e* **in** *examples* **do**

*O*  $\leftarrow$  NEURAL-NETWORK-OUTPUT(*network*, *e*)

*T*  $\leftarrow$  the observed output values from *e*

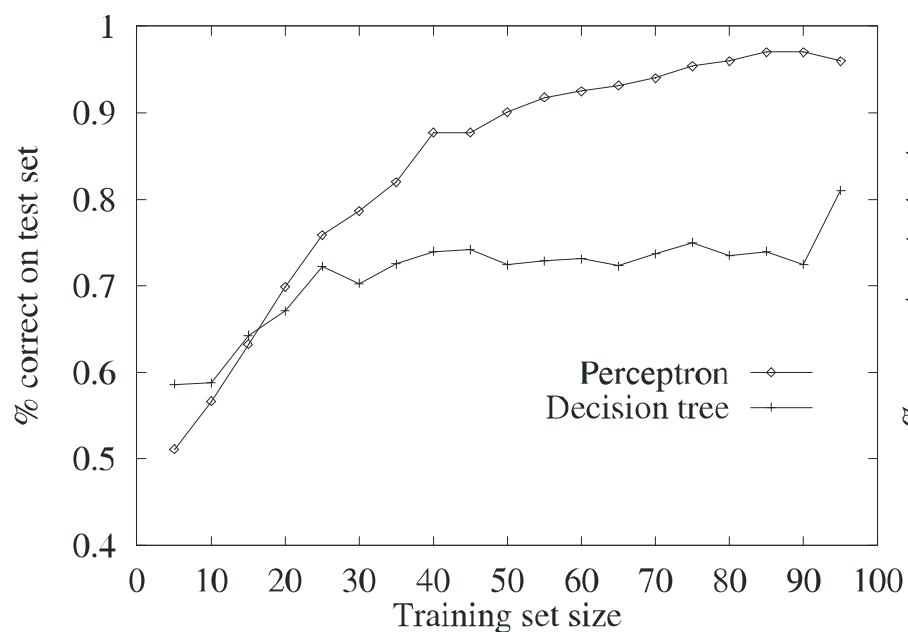
    update the weights in *network* based on *e*, *O*, and *T*

**end**

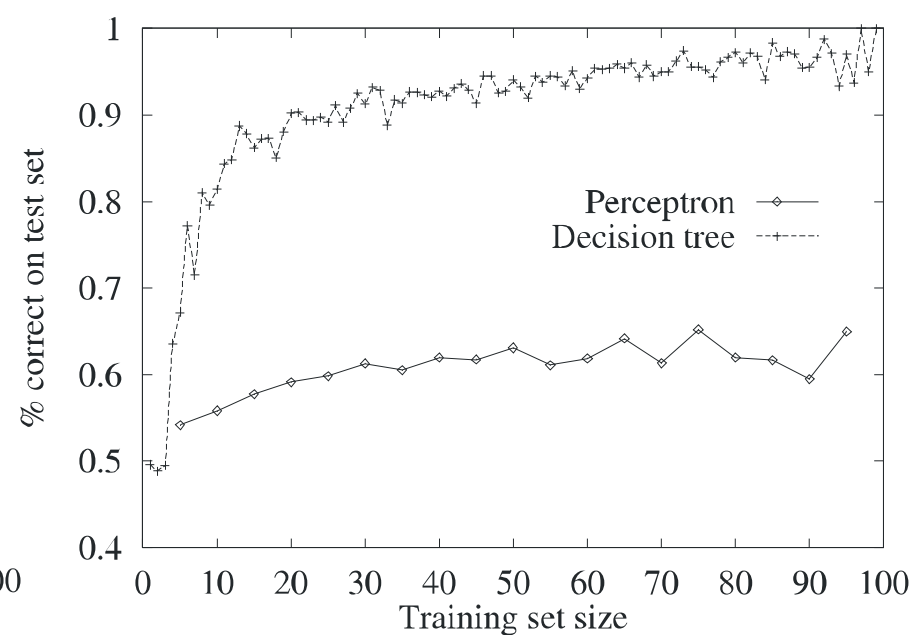
**until** all examples correctly predicted or stopping criterion is reached

**return** *network*

# Обука на НМ и дрвата на одлука



majority (11 bits)



restaurant



# Повеќеслојни невронски мрежи

- Multilayer feed-forward neural networks (Rosenblatt 1950s)
- Проблем – обуката
- Back-propagation алгоритам (1969, 1980)

# Повеќеслојни невронски мрежи

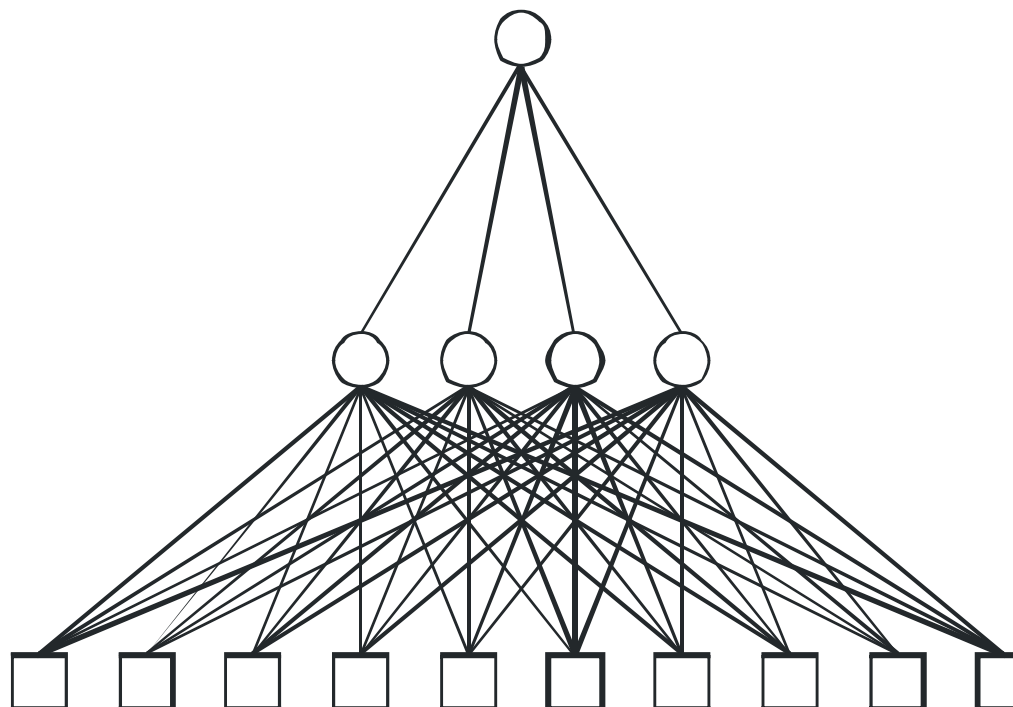
Output units  $O_i$

$W_{j,i}$

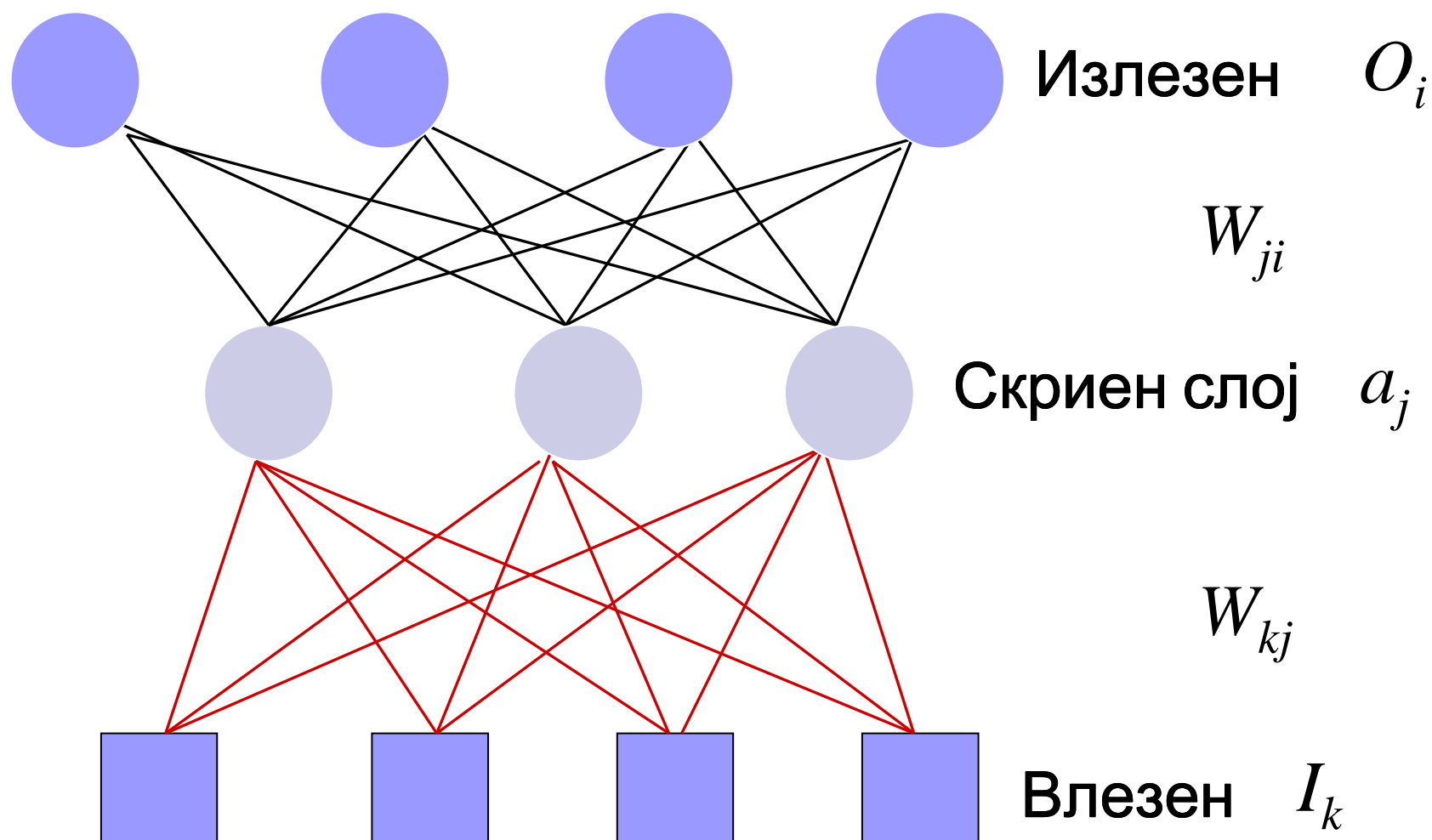
Hidden units  $a_j$

$W_{k,j}$

Input units  $I_k$



# Back-Propagation



# Грешка кај повеќеслојните НМ

- Да се подели одговорноста за грешката на соодветните тежини

$$Err_i = T_i - O_i$$

$$W_{j,i} \leftarrow W_{j,i} + \alpha \cdot a_j \cdot Err_i \cdot g'(in_i)$$

- по смена:  $\Delta_i = Err_i \cdot g'(in_i)$

$$W_{j,i} \leftarrow W_{j,i} + \alpha \cdot a_j \cdot \Delta_i$$

- скриениот јазол  $j$  е „одговорен“ за дел од грешката  $\Delta_i$  во секој од излезните јазли со кои е поврзан. Грешката  $\Delta_i$  се дели според јачината на врската помеѓу скриениот и излезниот јазел и се пропагираат наназад како грешки за невроните од скриеното ниво.

# Распространување (пропагација) на грешката

Правило за  
распространување

$$\Delta_j = g'(in_j) \sum_i W_{j,i} \Delta_i$$

Промена на тежините

Влез - скриен слој

$$W_{k,j} \leftarrow W_{k,j} + \alpha \cdot I_k \cdot \Delta_j$$

Алгоритам:

- Пресметај ја  $\Delta$  вредноста за излезните неврони според забележаната грешка
- Почнувајќи од излезното ниво повторувај за секое ниво на мрежата се до влезното ниво:
  - Пропагирај ги  $\Delta$  вредностите назад низ претходното ниво
  - Измени ги тежините помеѓу двете нивоа

# Back-propagation algorithm\*

**function** BACK-PROP-UPDATE(*network*, *examples*, *a*) **returns** a network with modified weights

**inputs:** *network*, a multilayer network

*examples*, a set of input/output pairs

$\alpha$ , the learning rate

**repeat**

**for each** *e* **in** *examples* **do**

*/\* Compute the output for this example \*/*

$\mathbf{O} \leftarrow \text{RUN-NETWORK}(\text{network}, \mathbf{I}^e)$

*/\* Compute the error and  $\Delta$  for units in the output layer \*/*

$\text{Err}^e \leftarrow \mathbf{T}^e - \mathbf{O}$

*/\* Update the weights leading to the output layer \*/*

$W_{j,i} \leftarrow W_{j,i} + \alpha \cdot a_j \cdot \text{Err}_i^e \cdot g'(in_i)$

**for each** subsequent layer **in** *network* **do**

*/\* Compute the error at each node \*/*

$\Delta_j \leftarrow g'(in_j) \sum_i W_{j,i} \Delta_i$

*/\* Update the weights leading into the layer \*/*

$W_{k,j} \leftarrow W_{k,j} + \alpha \cdot I_k \cdot \Delta_j$

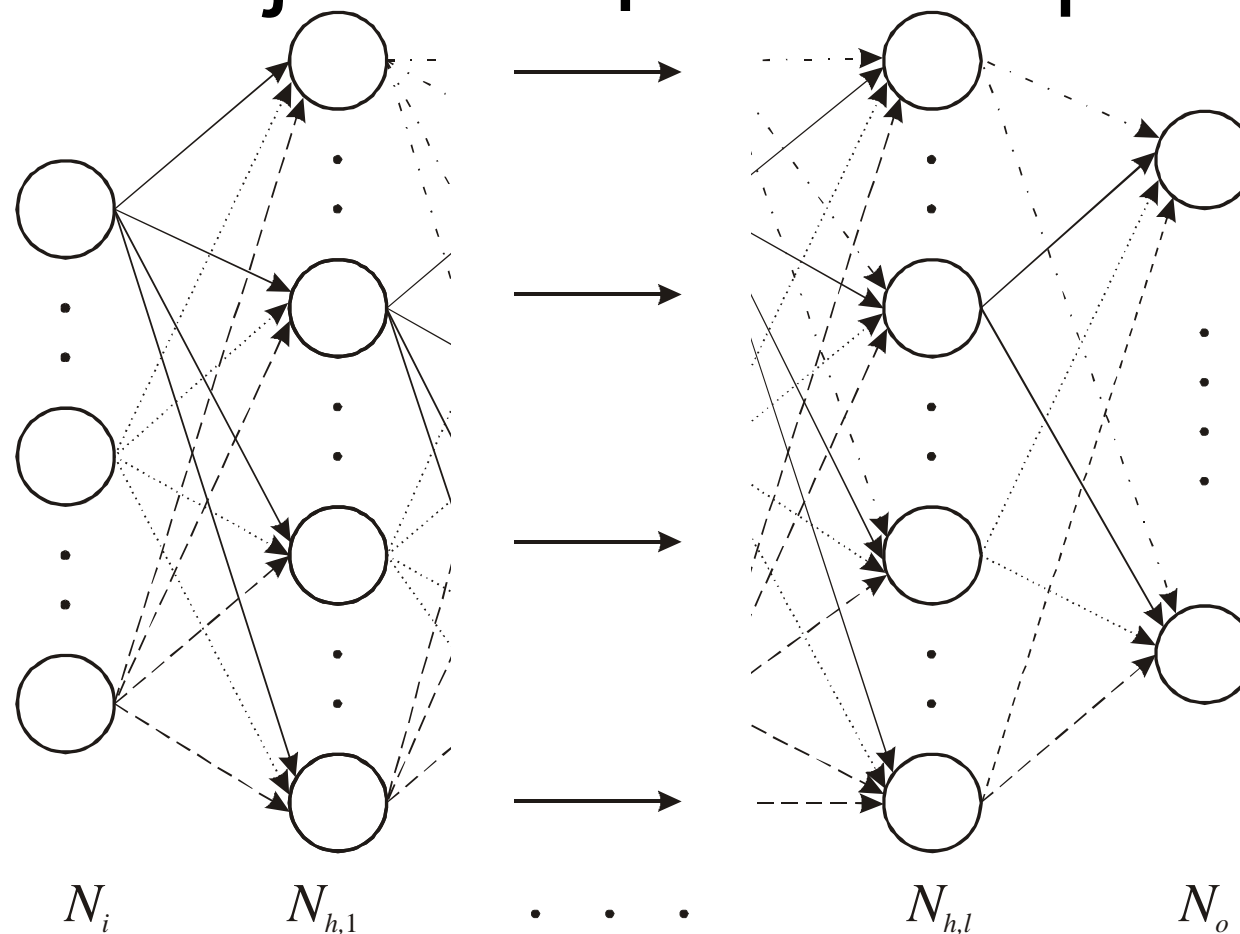
**end**

**end**

**until** *network* has converged

**return** *network*

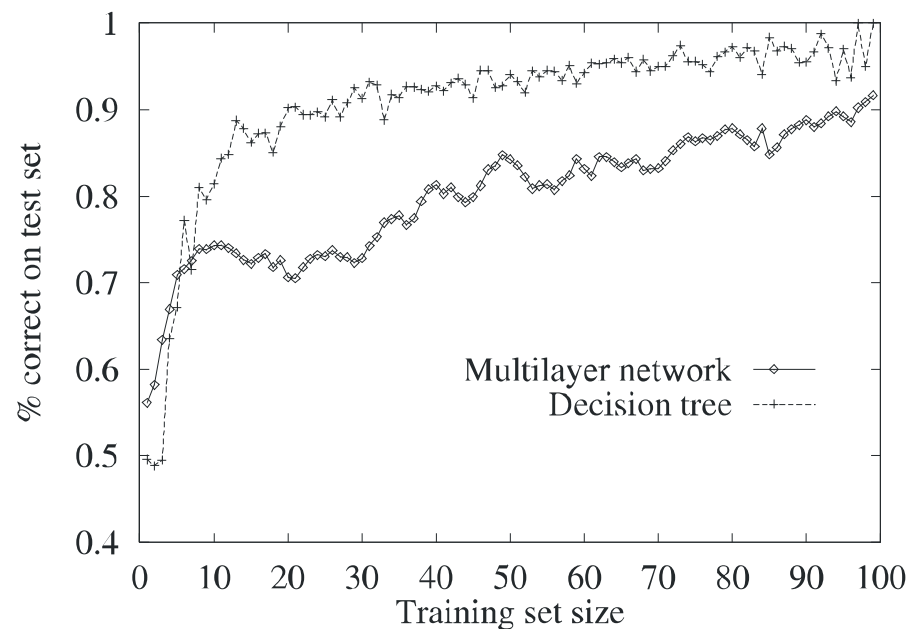
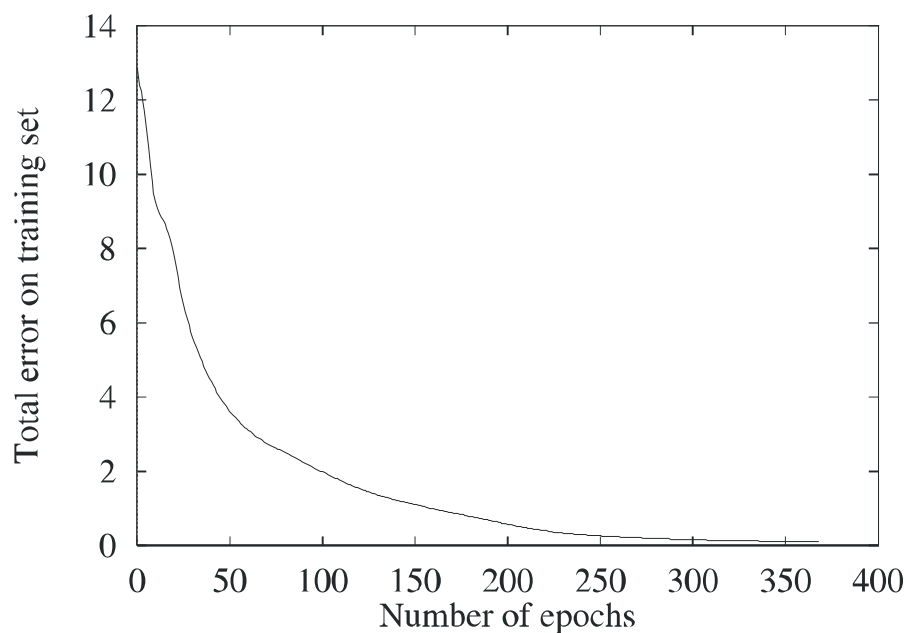
# Повеќеслојни невронски мрежи



Се покажало дека нема потреба да се оди со повеќе слоеви, туку истата пресметковна моќ и мемориски капацитет може да се постигне со зголемување на бројот на јазли во скриениот слој

# Грешка при обуката

- Невронска мрежа со две нивоа (10-4-1)  
за проблемот „ресторан“





# Карактеристики

- Експресивност
- Пресметковна ефикасност
- Генерализација
- Осетливост на шум (но има и робустни варијанти – отпорни на шум)
- Транспарентност? (за жал – не)

Примени...

# За што се добри невронските мрежи?

- Добри препознавачи на урнеци и робустни класификатори
- Добри при решавање на проблеми кои се премногу сложени за конвенционалните технологии
- Идеални решенија за мноштво проблеми како и за предвидувања



# Прашања?

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Илустрација на еден тип невронски мрежи

# Пример за една елементарна невронска мрежа

## ■ Околина:

- ☐ еден семафор резултатите ги покажува со помош на цифри претставени со мали сијалички во хоризонтала и вертикала
- ☐ сензорот кој ги чита цифрите е во состојба да открие дали светат сијаличките, но е ограничен на најмногу 5 кои се непрекинати.
- ☐ бројот на хоризонтални сијалички е 5, а на верикални 9.

## ■ Задача: Да се обучи една невронска мрежа за да може да ги препознае

- ☐ сите цифри од резултатот кога семафорот е исправен
- ☐ цифрите од резултатот ако некои сијалички се неисправни

# Процесни единици

- 10 различни објекти треба да се претстават бинарно, значи дека треба да се најдат барем 4 признака кои генерираат 10 различни бинарни броеви.
- Најпросто, оваа задача може да се реши и со 7 процесни единици, при што секоја процесна единица би била една од седумте линии од кои е составена цифрата.
- Но, може да се најдат и комбинациите од 5 сијалички кои се појавуваат на еден или најмногу два правци (ги има барем 18) и потоа да се изврши одделување на четирите најважни, со помош на генетските алгоритми.

# Нашиот рачен избор на процесните единици

## ■ Влезни единици:

- ☐ агол горе лево (5 го имаат, 5 го немаат)
- ☐ агол средина горе десно (5 го имаат, 5 го немаат)
- ☐ агол горе десно (7 го имаат, 3 го немаат)
- ☐ црта долу лево (4 ја имаат, 6 ја немаат)

## ■ Излезни единици:

- ☐ Бројот претставен со помош на три цифри кои имаат вредности 0, 1 или 2.

# Влезно-излезни парови

■ 0: 1 0 1 1	1 0 0
■ 1: 0 0 0 0	0 0 0
■ 2: 0 1 1 1	0 0 1
■ 3: 0 1 1 0	0 0 2
■ 4: 0 1 0 0	0 1 0
■ 5: 1 0 0 0	0 1 1
■ 6: 1 0 0 1	0 1 2
■ 7: 0 0 1 0	0 2 0
■ 8: 1 1 1 1	0 2 1
■ 9: 1 1 1 0	0 2 2



Зошто се измешани претставувањата на броевите со помош на бројниот систем со основа 3?

- Затоа што цифрата 1 е претставена со 4 нули, а тие со било која трансформација би преминале само во нули.
- За да се минимизира грешката (што е целта на обучувањето), цифрата 1 има влезно-излезна асоцијација составена само од нули.
- Според тоа, бројот кој е излез нумерички ќе се претстави како цифра на единиците при премин на следбеникот на бројот од системот со основа 3 во декадниот систем.

# Што е целта?

- Ако матрицата на влезот (10 редици, 4 колони) се помножи со тежинската матрица (3 редици, 4 колони), треба да се добие што е можно помала разлика меѓу излезот (тоа е производот на двете матрици) и очекуваната цел (10 редици, 3 колони)
- На ваков начин се дефинира систем од 10 равенки со 12 непознати, кој има:
  - ☐ бесконечно многу решенија (што не е еднозначно)
  - ☐ противречен (што не ни треба)

# Или, со матрици:

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \\ w_{41} & w_{42} & w_{43} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \\ 0 & 2 & 2 \end{bmatrix}$$

# Прва итерација: еднакви тежини

- Нека е матрицата  $W = \frac{1}{2} E_{4 \times 4}$ .
- Причината за изборот на  $\frac{1}{2}$  е што најголемата целна вредност е 2, а таа е збир на четири единици (нив ги има во влезот на цифрата 8).

# Излезна матрица

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} ,5 & ,5 & ,5 \\ ,5 & ,5 & ,5 \\ ,5 & ,5 & ,5 \\ ,5 & ,5 & ,5 \end{bmatrix} = \begin{bmatrix} 1,5 & 1,5 & 1,5 \\ 0 & 0 & 0 \\ 1,5 & 1,5 & 1,5 \\ 1,0 & 1,0 & 1,0 \\ ,5 & ,5 & ,5 \\ ,5 & ,5 & ,5 \\ 1 & 1 & 1 \\ ,5 & ,5 & ,5 \\ 2 & 2 & 2 \\ 1,5 & 1,5 & 1,5 \end{bmatrix}$$

# Разлика меѓу излезот и целта

$$\text{Output} = \begin{bmatrix} 1,5 & 1,5 & 1,5 \\ 0 & 0 & 0 \\ 1,5 & 1,5 & 1,5 \\ 1,0 & 1,0 & 1,0 \\ ,5 & ,5 & ,5 \\ ,5 & ,5 & ,5 \\ 1 & 1 & 1 \\ ,5 & ,5 & ,5 \\ 2 & 2 & 2 \\ 1,5 & 1,5 & 1,5 \end{bmatrix} \quad \text{Target} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 2 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 2 & 0 \\ 0 & 2 & 1 \\ 0 & 2 & 2 \end{bmatrix}$$

# Критериум за грешка

- Средно-квадратно отстапување меѓу излезот и целта
- Колона 1: 1,15
- Колона 2: 0,85
- Колона 3: 0,65

0,25	2,25	2,25
0	0	0
2,25	2,25	0,25
1	1	1
0,25	0,25	0,25
0,25	0,25	0,25
1	0	1
0,25	2,25	0,25
4	0	1
2,25	0,25	0,25

# Рачно нагодување на првата колона според Хебовиот алгоритам\*

- Најголема грешка има кај првата колона; нејзината вредност е 1,15
- Нагодувањето се врши за сите три колони истовремено, но во илустрацијата се задржуваме само на првата, бидејќи пристапот е сосема идентичен.
- Доколку сакате, работете рачно за сите три колони, но тоа полесно ќе ви го реши MatLab.



# Како да се нагоди првата колона од тежинската матрица?\*

- Како прво, се дефинира вредноста за која ќе се врши промената. Таа се вика рата за учење (learning rate). Да земеме дека таа изнесува  $\frac{1}{4}$ .
- Потоа, користејќи го Хебовиот пристап, се одредува кој од четирите тежински фактори треба да се зголеми, а кој да се намали.
- Проценуваме дека цифрите 2, 9, а посебно 8 имаат голема грешка. Значи, тежинските фактори што се однесуваат на нивните влезни низи треба да се намалат. Во случајов, тоа се вториот и третиот тежински фактор. Првиот и четвртиот нека останат непроменети.

# Новиот излез и очекуваната цел\*

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} ,5 \\ ,25 \\ ,25 \\ ,5 \end{bmatrix} = \begin{bmatrix} 1,25 \\ 0 \\ 1 \\ ,5 \\ ,25 \\ ,5 \\ 1 \\ ,25 \\ 1,5 \\ 1 \end{bmatrix} \quad \text{Target} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

# Следна епоха (или, итерација)\*

- Доколку како излезна функција се земе Хевизајдовата функција од излезот, тогаш по првата итерација, проблематични остануваат грешките на цифрите 2, 6, 8 и 9.
- Во нивната грешка, удел имаат трите десни тежини. Повторно се намалуваат за истата рата, т.е.  $\frac{1}{4}$ . За да не се намали точната вредност на излезот за цифрата 0, за истата рата се зголемува првиот тежински фактор.
- Ако Хевизајдовата функција има праг 1, тогаш сите излезни вредности под 1 се 0.

# Излезот и целта по втората епоха\*

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} ,75 \\ 0 \\ 0 \\ ,25 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ ,25 \\ 0 \\ 0 \\ ,75 \\ 1 \\ 0 \\ 1 \\ 1 \end{bmatrix} \quad \text{Target} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

# Пред почетокот на третата епоха\*

- Излезот за цифрите 6, 8 и 9 е неприфатливо голем, па затоа одново се намалуваат трите десни тежини и повторно се зголемува првата.
- Ратата останува  $\frac{1}{4}$ .

# Излезот и целта по третата епоха\*

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} 1 \\ -0,25 \\ -0,25 \\ 0 \end{bmatrix} = \begin{bmatrix} 0,75 \\ 0 \\ -0,5 \\ -0,5 \\ -0,25 \\ 1 \\ 1 \\ 0 \\ 0,5 \\ 0,5 \end{bmatrix} \quad \text{Target} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

# Влез во следна епоха\*

- Првиот тежински фактор е неоправдано фаворизиран, посебно затоа што тој придонесува за високата грешка на цифрите 5 и 6, но третата и четвртата тежина може да се зголемат.

# Излезот и целта по четвртата епоха\*

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} ,75 \\ -,25 \\ 0 \\ ,25 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ -,25 \\ -,25 \\ ,75 \\ 1 \\ 0 \\ ,75 \\ ,5 \end{bmatrix} \quad \text{Target} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



# Следна епоха\*

- Повторно се намалува само првата тежина (за да ја намали вредноста на цифрата 6), а се зголемува уделот на третата, за да се зајакне цифрата 0.

# Излезот и целта по петтата епоха\*

$$\begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} * \begin{bmatrix} ,5 \\ -,25 \\ ,25 \\ ,25 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ ,25 \\ 0 \\ -,25 \\ ,5 \\ ,75 \\ ,25 \\ ,75 \\ ,5 \end{bmatrix} \quad \text{Target} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

# Постапката е завршена

- Со оглед на тоа што излезната функција е Хевизајдова, се доаѓа до идеално обучување со множеството за обука.
- Ваков резултат не може да се добие, но грешката треба да се намали што е можно повеќе.
- Притоа, ратата за обучување може да има динамичка вредност, т.е. да се менува во тек на следните епохи.
- Алгоритмите за учење се далеку покомплексни, но оваа крајно упростена стратегија е извонредна за да се сфати како работат НМ.

# Валидација

- Во нашиов пример, нема простор за валидација, бидејќи множеството за обука е многу мало.
- Сепак, една 10fold вкрстена исправност (научи 9 цифри и провери дали се препознава десеттата), има смисол.

# Тестирање

- По завршувањето на обучувањето на сите 12 тежински фактори, претстои тестирање.
- Тестирањето во овој пример би се однесувало на проверка на тоа колку е способна обучената мрежа да ги препознае цифрите од семафорот во кои некои од сијаличките се неисправни.
- Проверете колкава е моќта на нашата обучена мрежа.



# Прашања?



ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Еволутивни алгоритми

## Генетски алгоритми

# Зошто еволутивни алгоритми?

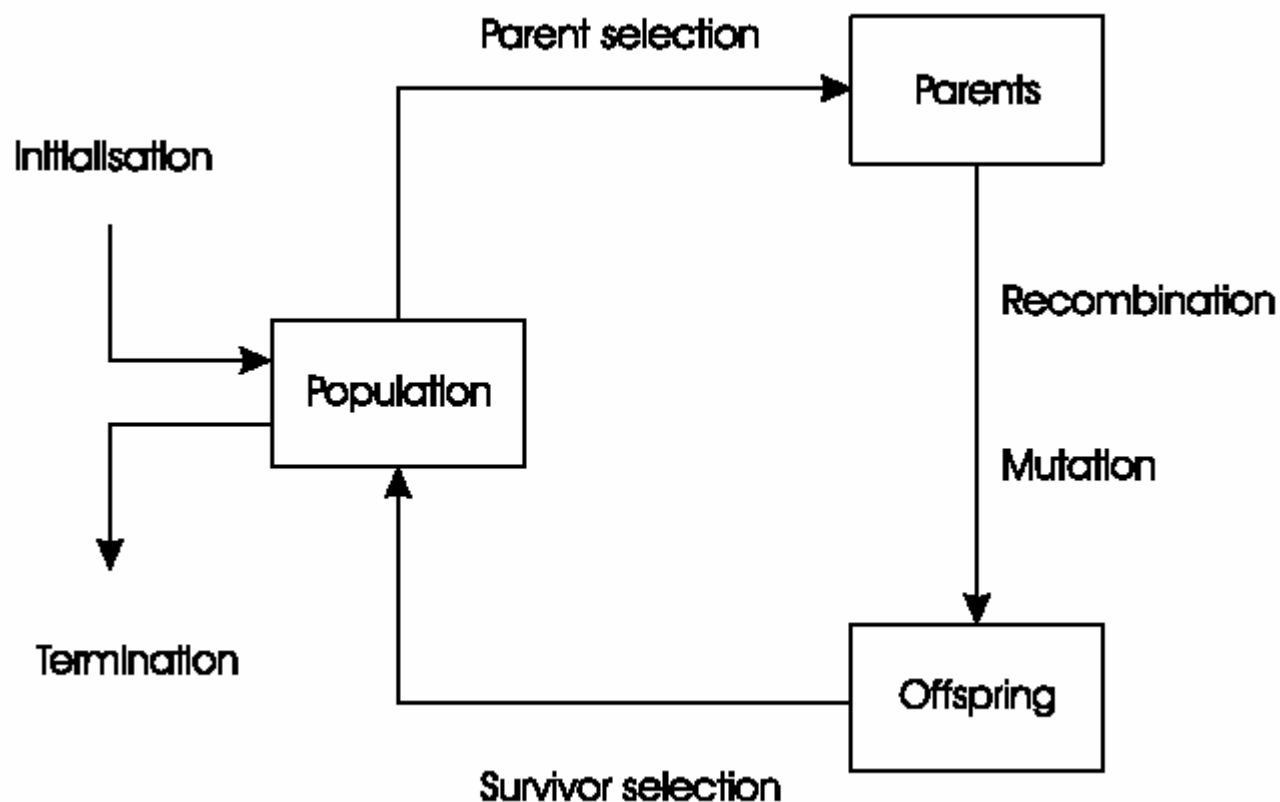
- Во реалноста постојат многу популации кои живеат во услови на недоволно ресурси
- Натпреварот за оскудните ресурси е причина за селекција на оние единки кои се поподготвени од останатите да се адаптираат на средината
- Избраните единки служат како семе од кое се генерираат нови единки преку вкрстување и мутации
- Новите единки ја докажуваат својата способност да се справат со средината и да се натпреваруваат со останатите за да преживеат
- Со тек на време, природната селекција резултира со популација која е посоодветна на условите и средината



# Својства на еволутивните алгоритми

- Припаѓаат на категоријата „генерирај и тестирај“ алгоритми
- Стохастичко- (веројатносно / пробабилистичко) популациски засновани алгоритми
- Со промена на операторите за реструктурирање (вкрстување и мутација) се создава доволна разноликост, а со тоа и можност за иновации
- Селекцијата ја намалува разноликоста и делува како сила која го форсира квалитетот.

# Општ приказ на еволутивните алгоритми



# Генетски алгоритми

- ГА се еволутивни алгоритми (инспирирани од биологијата)
  - наследство (inheritance)
  - избор или селекција (selection)
  - вкрстување (crossover / recombination)
  - мутација (mutation)
- Наоѓаат примена во: компјутерските науки, инженерството, економијата, хемијата, физиката, математиката, ...

# Генетски алгоритми

- Генетските алгоритми претставуваат веројатносни (probabilistic) алгоритми за пребарување кои:
  - Итеративно го трансформираат множеството (т.н. популација) од математички објекти (вообичаено низи од битови со фиксна должина)
  - За секој од тие објекти е придружена функција за проценка на „успешноста“ (fitness) во нова популација на објекти наследници, употребувајќи го Дарвиновиот принцип на природна селекција и употребувајќи операции аналогни на природните генетски појави на вкрстување и мутација.
- Техника за наоѓање на точното или приближно решение во проблеми на оптимизација или пребарување користејќи глобална хеуристика за пребарување.

# Имплементација на ГА (1)

- Се имплементираат како компјутерска симулација во која популација на апстрактни репрезентации (chromosomes, genotype or genome) на кандидатите решенија (individuals, creatures, or phenotypes) на дадениот оптимизациски проблем, еволуираат кон подобри решенија
- Вообичаено решенијата се претставени со низа од битови

# Имплементација на ГА (1)

- Еволуцијата почнува со популација на случајно генерирани индивидуи и се одвива во генерации.
- Во секоја генерација прво се проценува успешноста односно приспособеноста (fitness) на секоја од индивидуите
- Потоа случајно се избираат одреден број на индивидуи според нивната приспособеност
- Тие се модифицираат (вкрстуваат и мутираат) за да создадат нова популација, која понатаму се користи во следната итерација на алгоритмот.

# Претставување

- За да се имплементира ГА потребно е да се дефинираат:
  - генетска репрезентација на просторот на решенија
    - низи од битови, реални вредности, стебла, ...
  - функција на приспособеност (fitness function)

# Genetic Algorithm

**function** GENETIC-ALGORITHM(*population*, FITNESS-FN) **returns** an individual

**inputs:** *population*, a set of individuals

FITNESS-FN, a function that measures the fitness of an individual

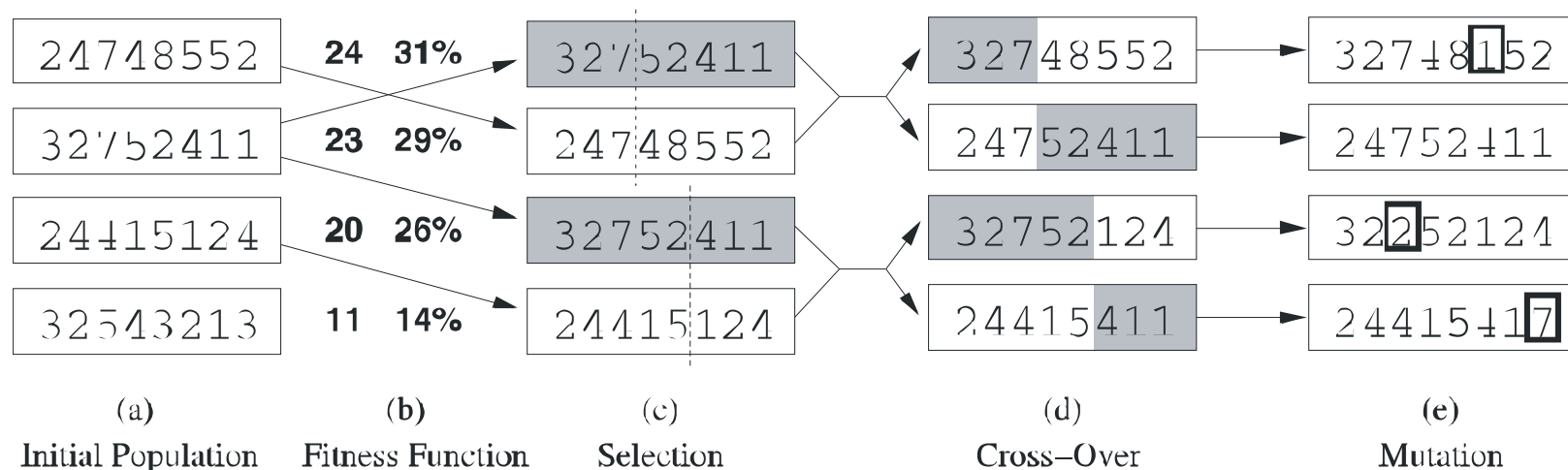
**repeat**

*parents*  $\leftarrow$  SELECTION(*population*, FITNESS-FN)

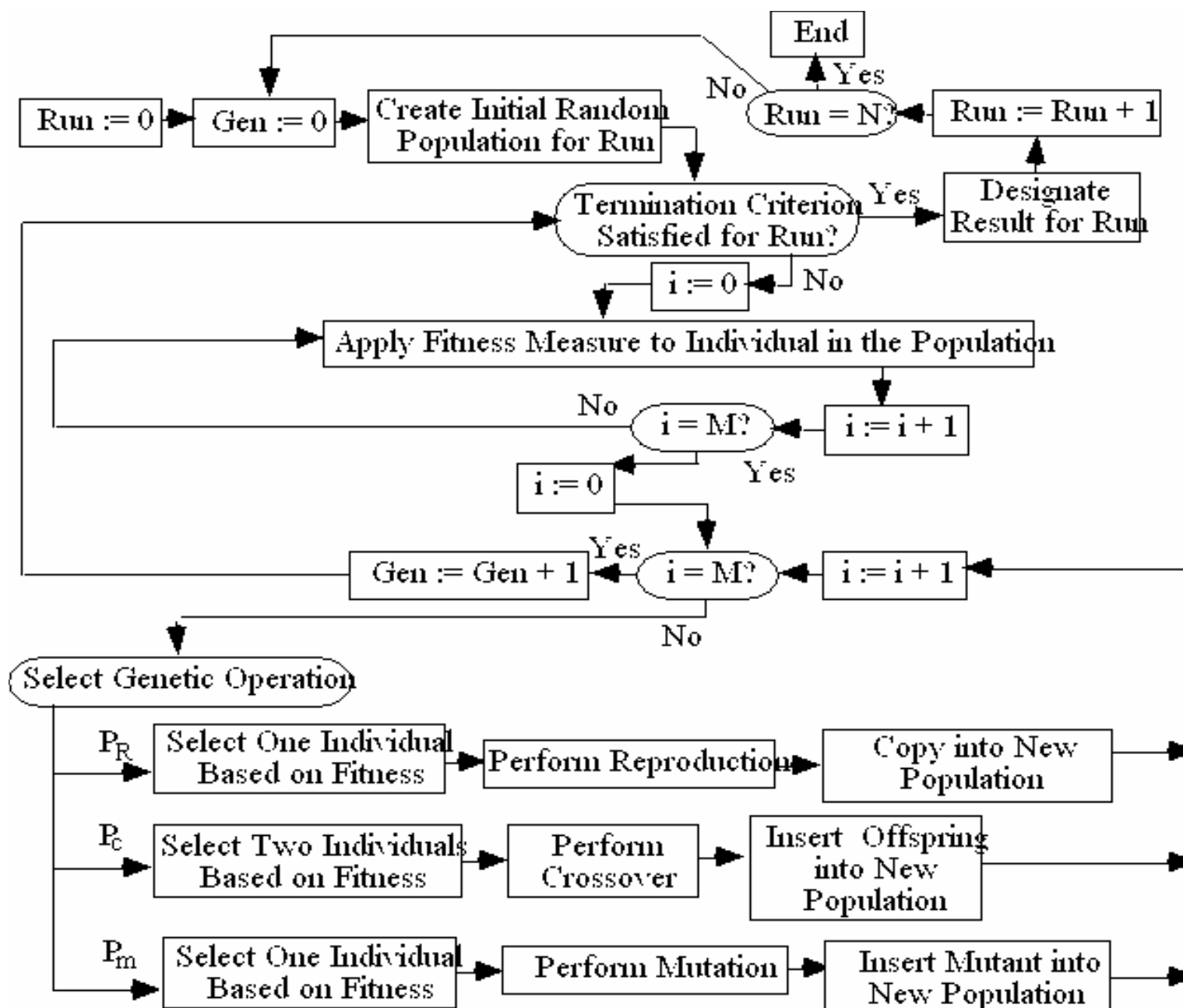
*population*  $\leftarrow$  REPRODUCTION(*parents*)

**until** some individual is fit enough

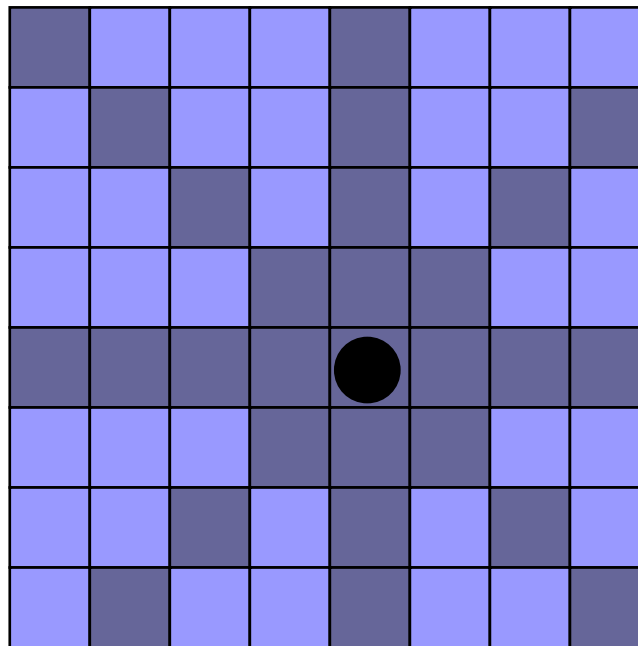
**return** the best individual in *population*, according to FITNESS-FN







# Проблем на 8 кралици (1)

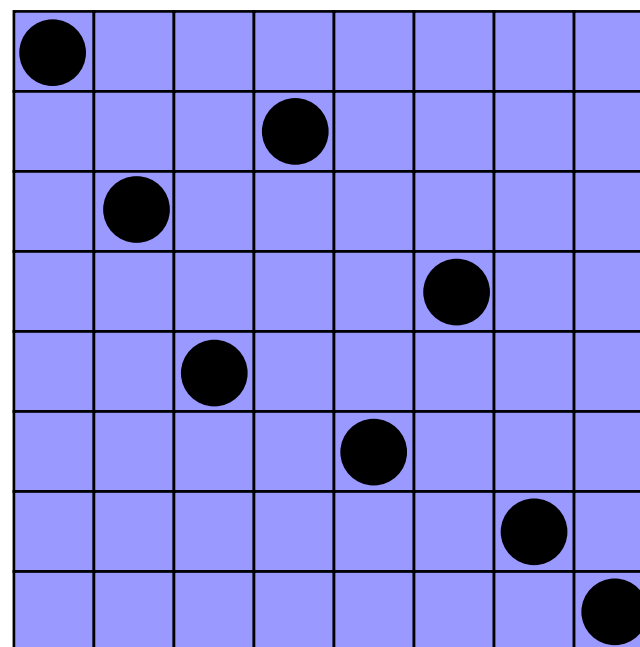


Постави 8 кралици на табла 8x8 на начин кој не им дозволува да се нападнат

# Проблем на 8 кралици (2)

Фенотип:

Конфигурација на табла



Генотип:

Пермутација на цифрите  
од 1 до 8

1	3	5	2	6	4	7	8
---	---	---	---	---	---	---	---



пресликување

# Проблем на 8 кралици (3)

- Казна за една кралица:  
Број на кралици кои може да ги матира.
- Казна за целата конфигурациска состојба:  
Сума на казните за сите кралици.
- Напомена: казната треба да се минимизира
- Оценка на целата конфигурација:  
Инверзната казна да се максимизира

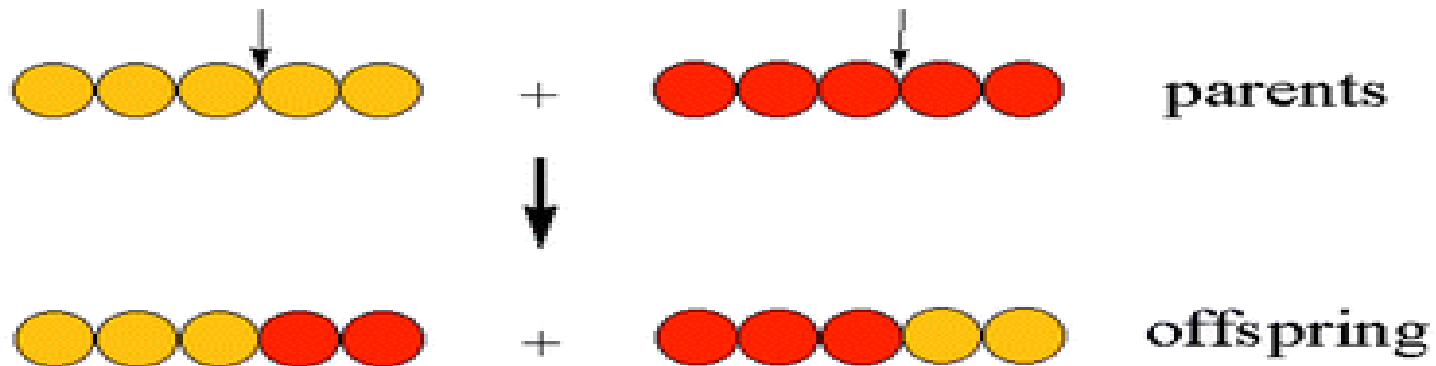
# Проблем на 8 кралици (4)

- Мутации - мали промени во една пермутација
  - Пр. Промена на местата на две произволно одбрани кралици

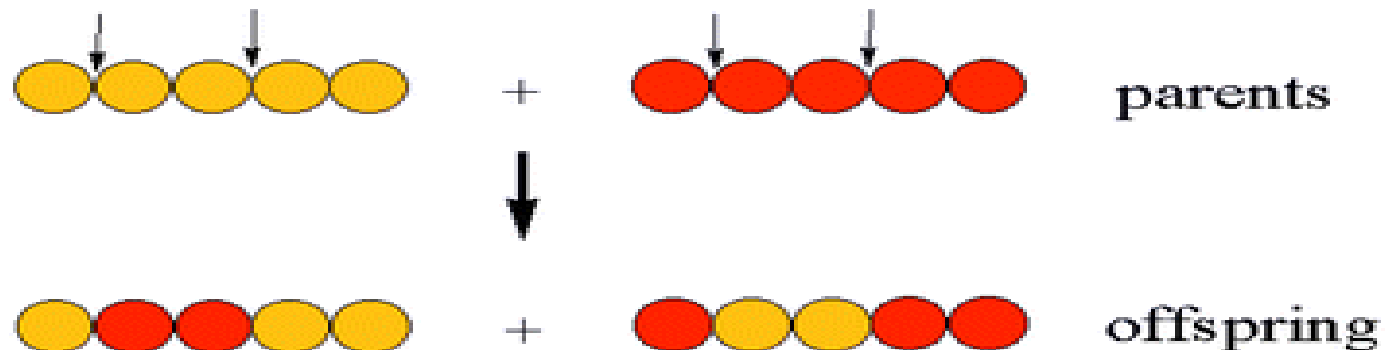


# Вкрстување со една и две точки

(a)

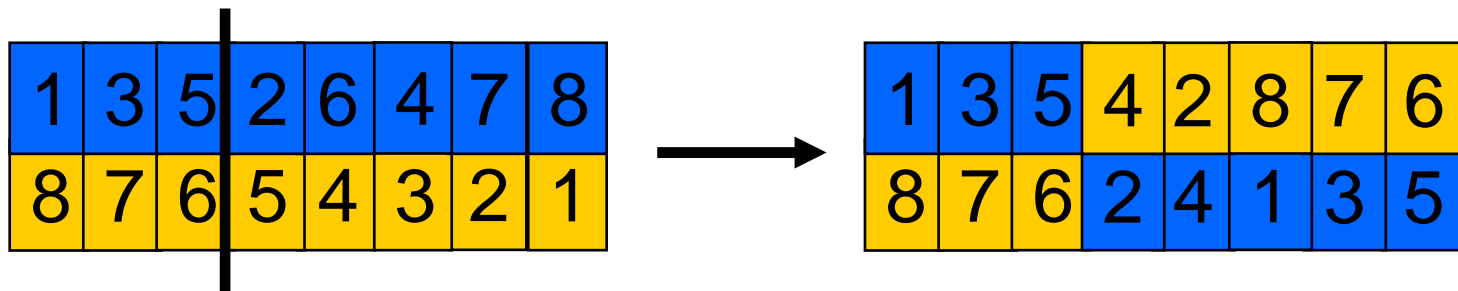


(b)



# Проблем на 8 кралици (5)

- Вкрстување - комбинирање на две пермутации во нова пермутација
  - случаен избор на точка за вкрстување
  - копирај ги првите делови од децата
  - креирај ги вторите делови со замена од другиот родител
    - со ист редослед како кај родителите
    - почнувајќи од точката на вкрстување
    - прескокнувајќи ги деловите кои се веќе наследени



# Проблем на 8 кралици (6)

- Селекција на родителите
  - Избери 5 родители и селектирај два кои учествуваат во вкрстувањето
- Селекција на оној кој ќе опстане
  - Кога ќе го внесуваш новиот член во популацијата, избери друг член кој ќе биде заменет
    - Подреди ја популацијата во опаѓачки редослед според функцијата за приспособеност
    - Нумерирај ја листата од оној со најголема оценка кон оној со најмалата
    - Замени ја првата единка со оценка помала од новиот член

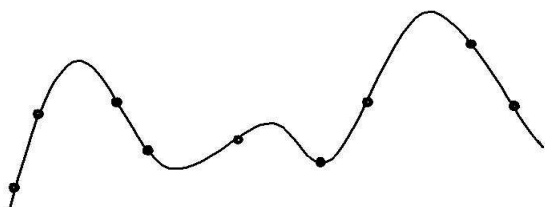


# Проблем на 8 кралици (7)

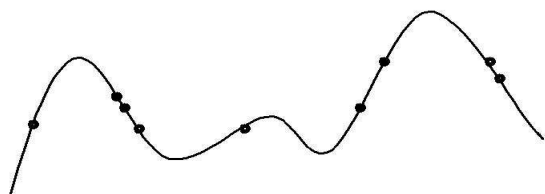
Representation	Permutations
Recombination	“Cut-and-crossfill” crossover
Recombination probability	100%
Mutation	Swap
Mutation probability	80%
Parent selection	Best 2 out of random 5
Survival selection	Replace worst
Population size	100
Number of Offspring	2
Initialisation	Random
Termination condition	Solution or 10,000 fitness evaluation

Ова е само една варијанта за  
избор на оператори и параметри

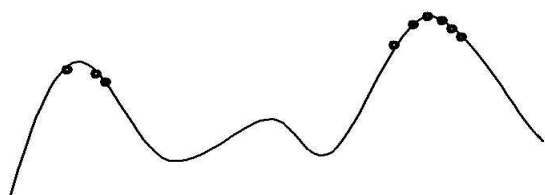
# Однесување на ЕА



Почетна фаза:  
Квази-случајна дистрибуција на популацијата

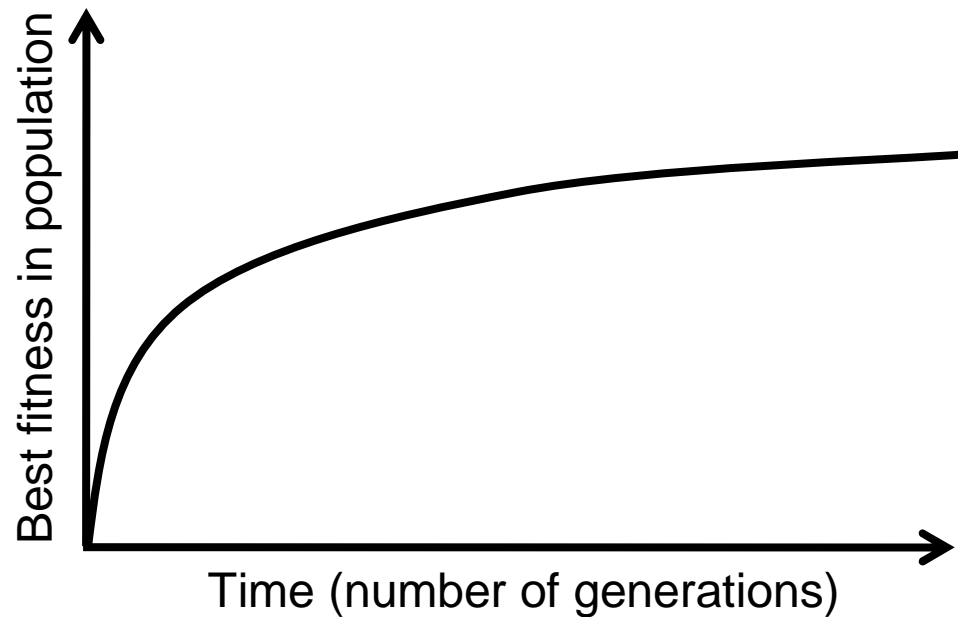


Средна фаза:  
Популација концентрирана околу падините

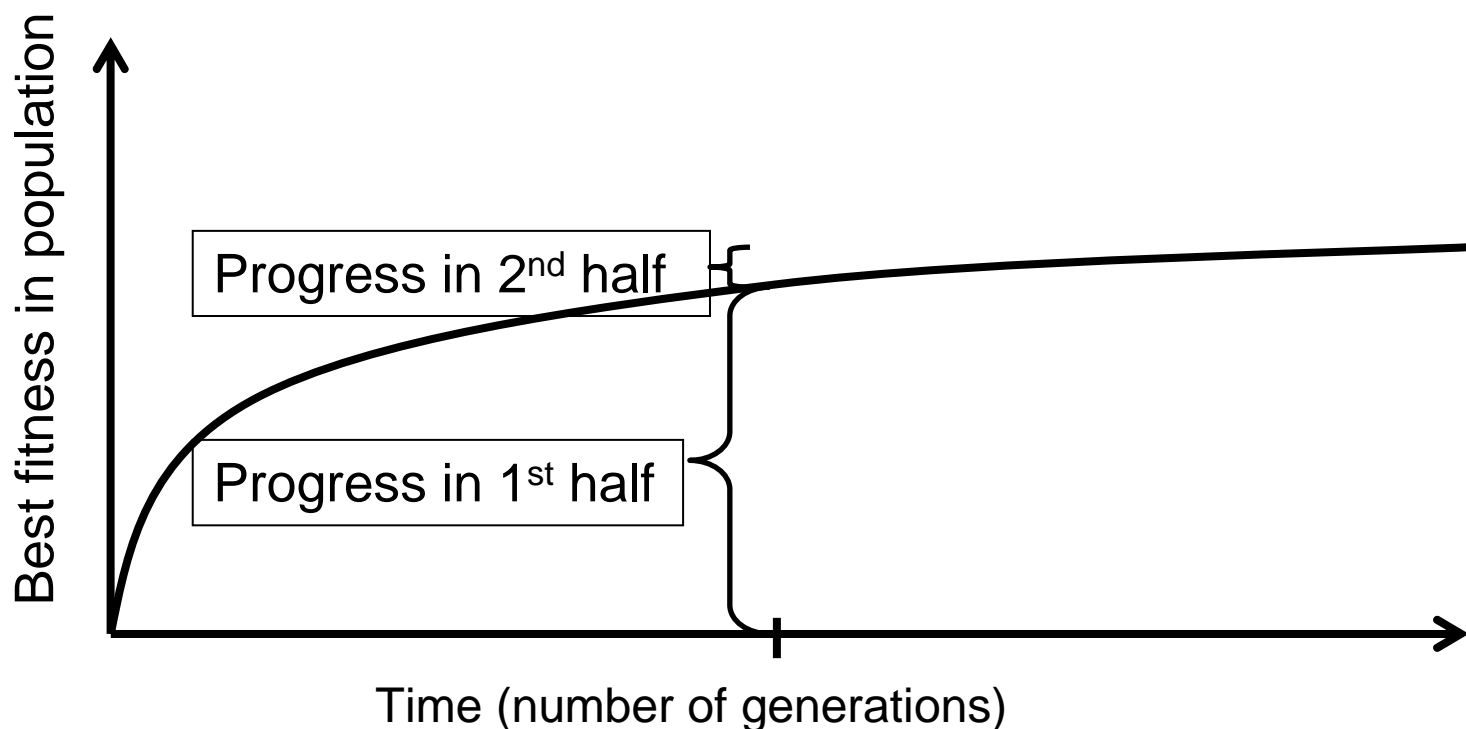


Доцна фаза:  
Популација концентрирана околу врвовите

# Прогресија на приспособувањето - fitness

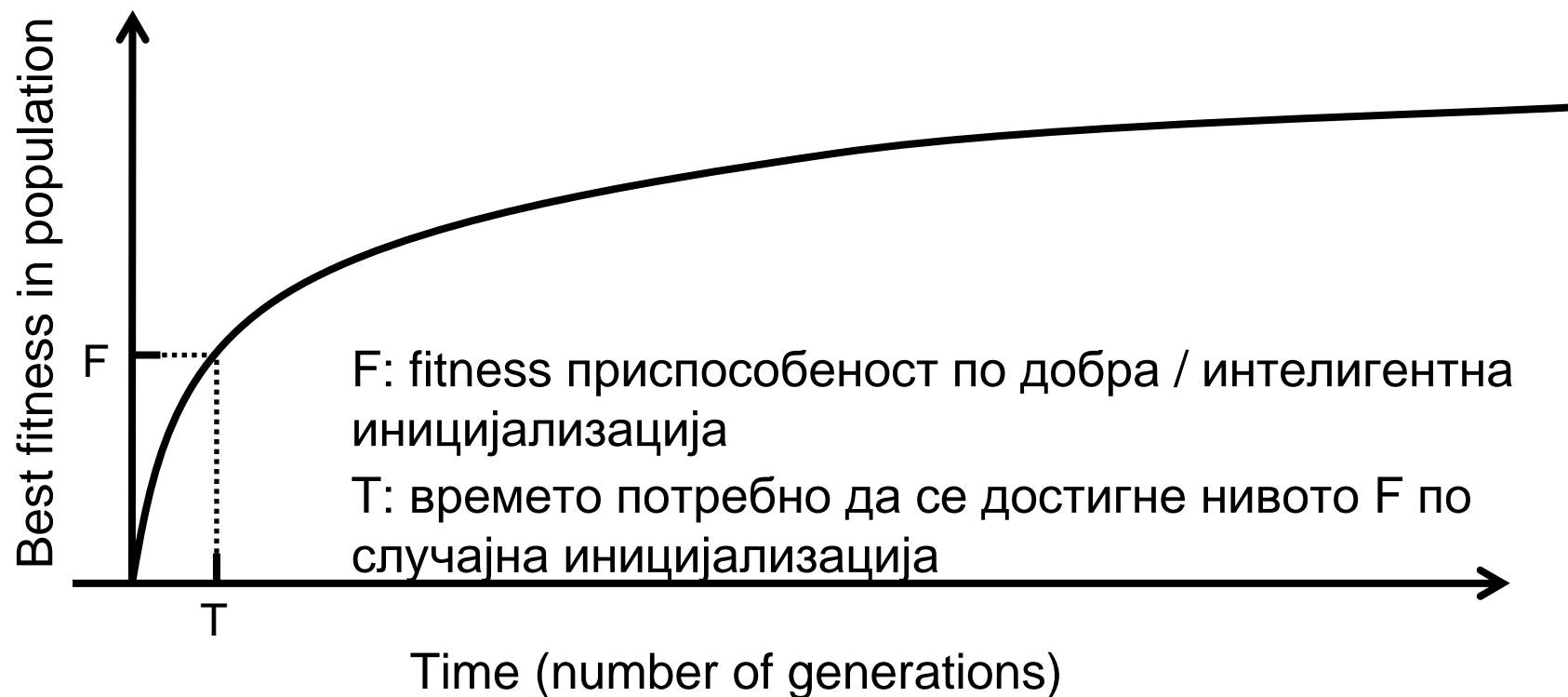


# Колку долги треба да бидат циклусите (run)?



- Одговорот зависи од тоа:
  - Колкаво подобрување се очекува во последните моменти од прогресот / приспособувањето
  - Понекогаш е подобро да имате пократки циклуси

## Колку е битно да се почне со добра иницијализација?



- Одговорот зависи од тоа:
  - Дали постојат добри / интелегентни почетни решенија?
  - Посебно внимание (хибридизација)

# Веројатносна селекција врз основа на приспособеноста

- Се преферираат подобрите индивидуи
- Не се одбира секогаш најдобриот
- Не мора секогаш се да отфрлуваат најлошите единици
- Не се гарантира ништо
- Спој на алчни и авантуростички истражувања
- Сличност со симулираното калење (simulated annealing)



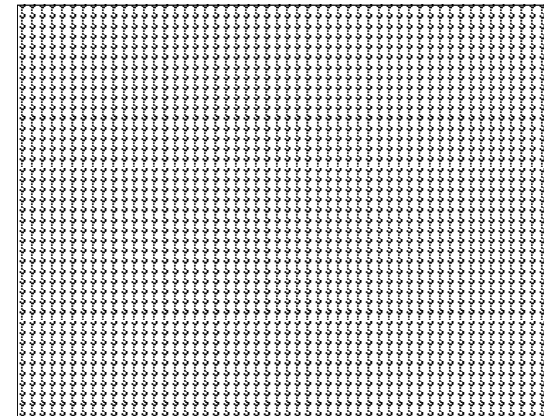
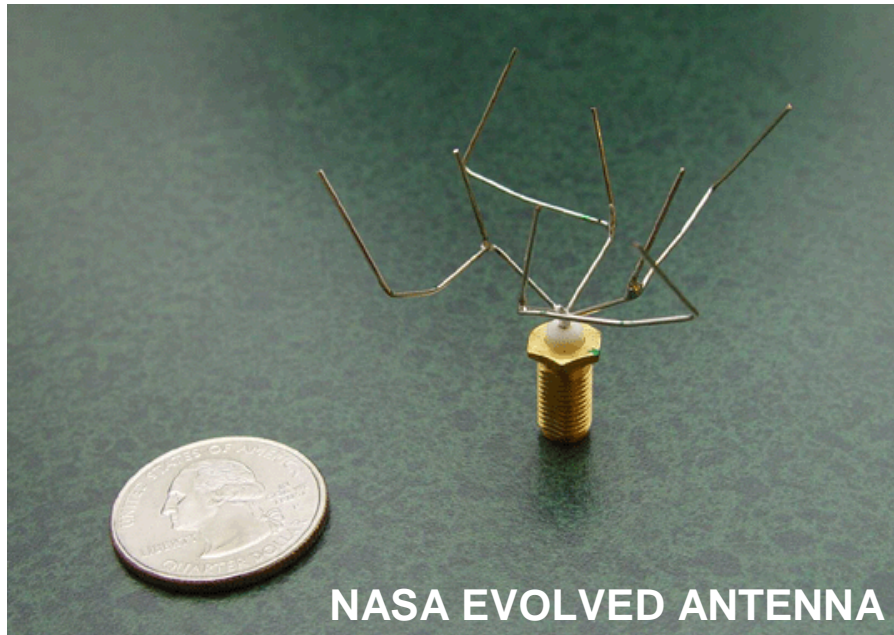
# Варијанти

- Иницијализација
- Селекција (fitness-based)
  - ☐ избор само на најдобрите
  - ☐ случаен избор
    - roulette wheel selection
    - tournament selection
- Репродукција
  - ☐ вкрстување (еднокатна, повеќекратна)
  - ☐ мутација
  - ☐ елитизам
- Прекинување
  - ☐ пронајдено задоволувачки добро решение
  - ☐ предодреден број генерации
  - ☐ се добило плато во fitness функцијата на популацијата

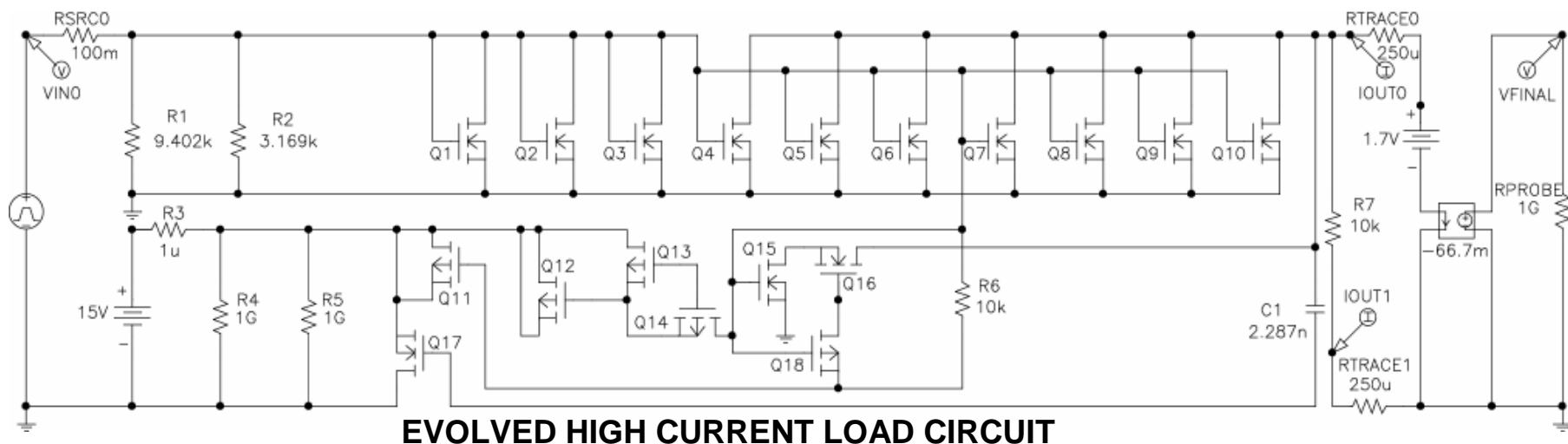


<http://cs.felk.cvut.cz/~xobitko/ga/>

# 3 POST-2000 PATENTED INVENTIONS



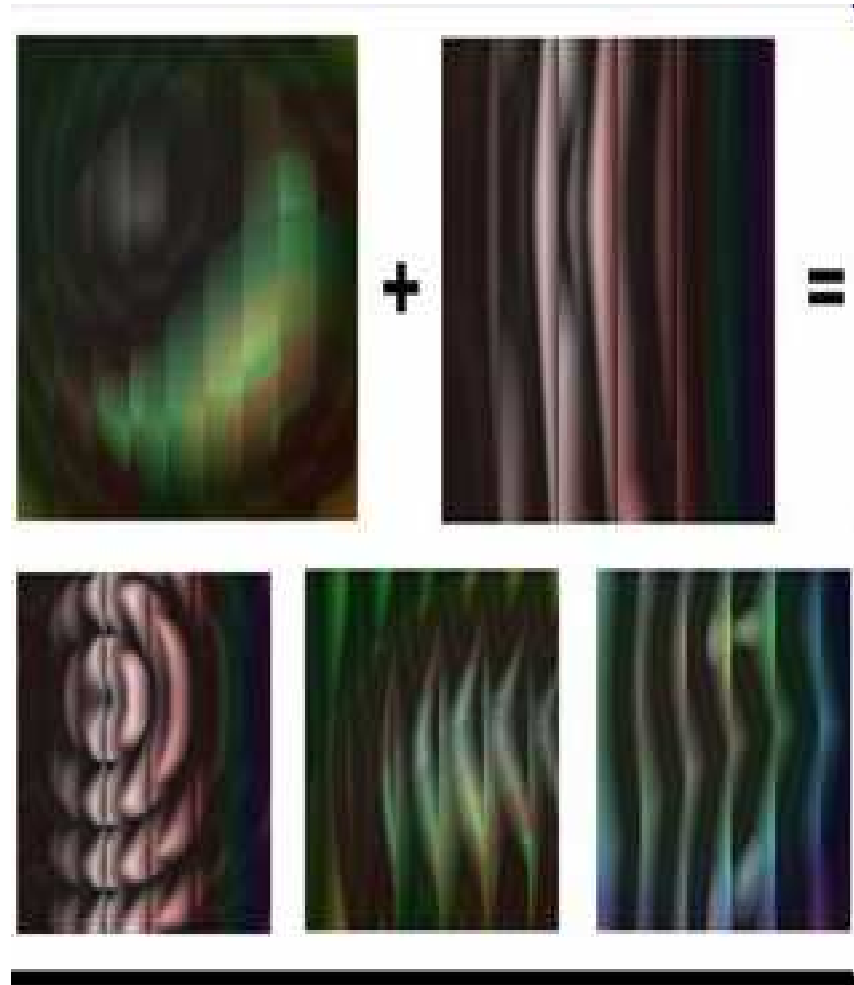
**REGISTER-CONTROLLED CAPACITOR CIRCUIT**





# Компјутерска креативност?

- Two portrait programs are mated together showing merged strategies of the offspring. Since the genes of each portrait can be saved, it is possible to re-combine (marry) and re-evolve any of the art works in new variants
- [www.darwinsgaze.com](http://www.darwinsgaze.com)



# Користена литература

- Artificial Intelligence, A Modern Approach  
2nd edition, Russel and Norvig
- Artificial Intelligence, A New Synthesis, Nils J. Nilsson
- Божиновски С., Вештачката интелигенција,  
Гоцмар, Скопје, 1994



# Прашања?

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Експертни системи

# Дефиниција

- Експертните системи се софтверски системи со чија помош се имитира (симулира) однесувањето (performance) на експертот при решавањето на проблемите од еден проблемски тип за коишто експертот е високо стручен.

# Зошто се значајни експертните системи (ЕС)

- Експертните системи се еден од најголемите успеси на вештачката интелигенција со висока практична примена.
- Тие поседуваат, но и стекнуваат сопствена база знаење и врз основа на правилата за одлучување што им се вградени донесуваат суд во врска со проблемите за кои се “стручни”.

# Својства на ЕС

- Користат огромни база знаење и расудуваат во доменот на знаењето.
- Употребуваат методи својствени за доменот на расудување.
- Работните карактеристики им се соодветни на дејноста на експертите од таа област.
- За знаењата коишто ги поседуваат и за начинот на којшто ги донесуваат одлуките даваат објаснување.
- Отворени се за нови информации и знаења од областа, т.е. во веќе постоечката структура може да вградуваат нови искуства.

# Како се овозможени овие својства?

(Образложение на претходниот слајд)

- Знаењето е претставено нумерички и симболички.
- Методите се алгоритамски и хеуристички.
- Работните карактеристики се соодветни на операционите истражувања, инженерството на деловните процеси и управувањето.
- Транспарентноста се обезбедува преку интерактивни дијалози.
- Прилагодливоста (најчесто) ја темелат врз компонентите на учење.



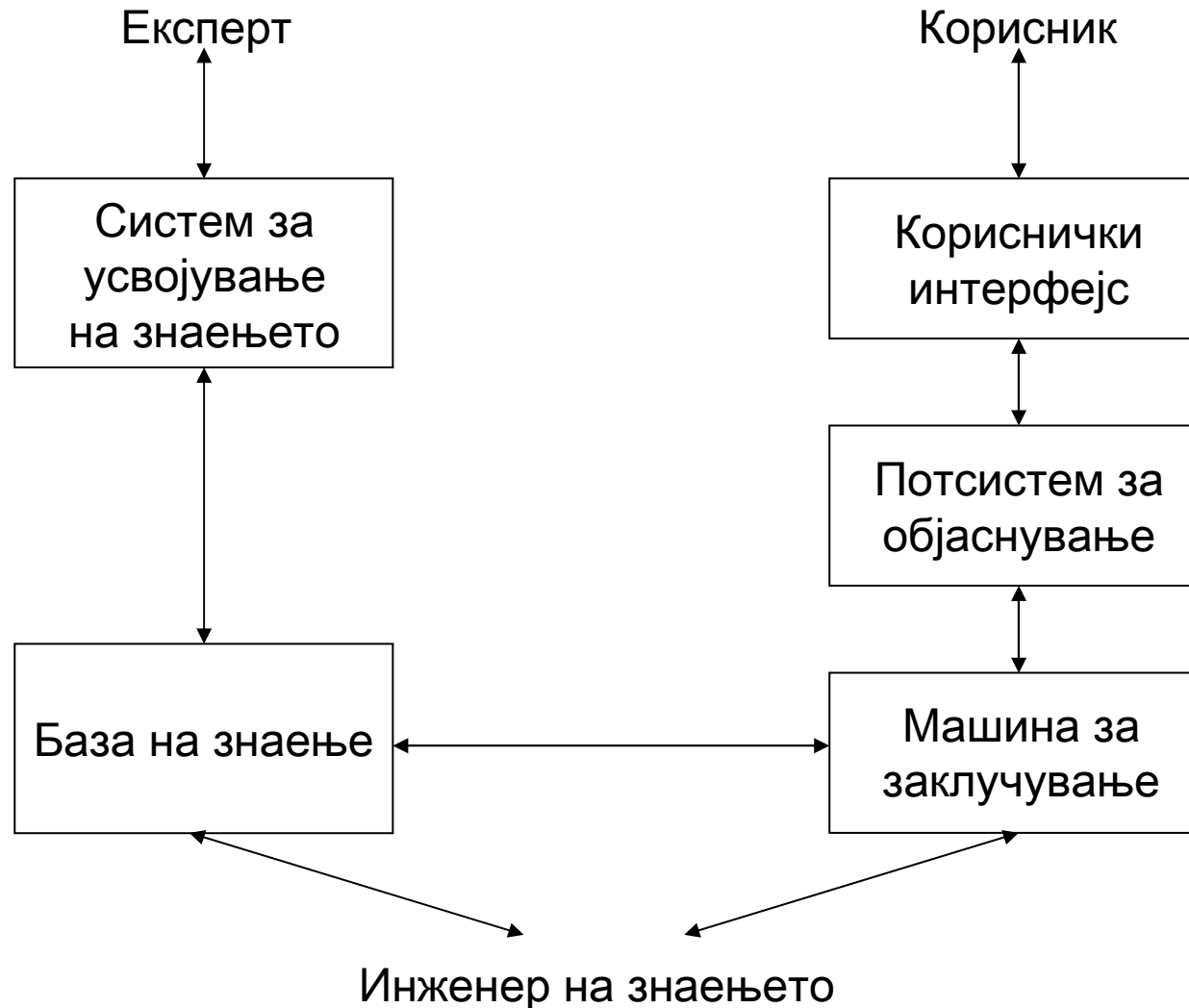
# Првиот експертен систем

- Heuristic Dendral создаден од Edward Feigenbaum и неговата група од Stanford University
- Со негова помош се одредува структурата на молекулата зададена со:
  - а) нејзината атомска формула и
  - б) масениот спектрограм.

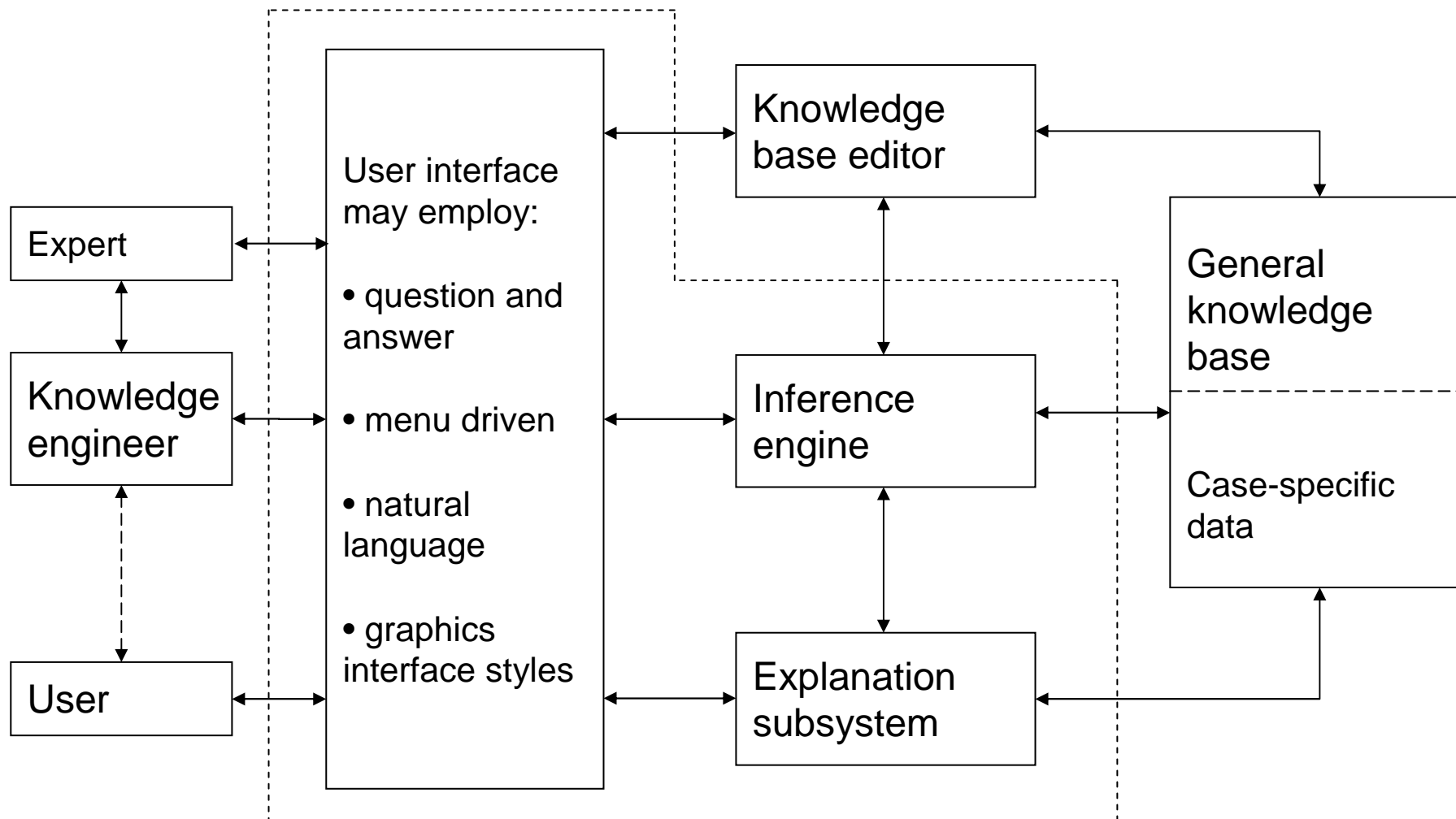
# Каде се применуваат ЕС?

- Медицинска дијагностика (MYCIN '75, EMYCIN '80, INTERNIST, ONCOSYN, ATTENDING )
- Електронски дизајн (R1 '82)
- Научна анализа (PROSPECTOR '79)
- Анализа на податоци за истражувањата на нафта (DipMeter Advisor)
- Финансиска анализа (MQL-4)
- Градба на експертни системи (LogicNets со/за NASA)

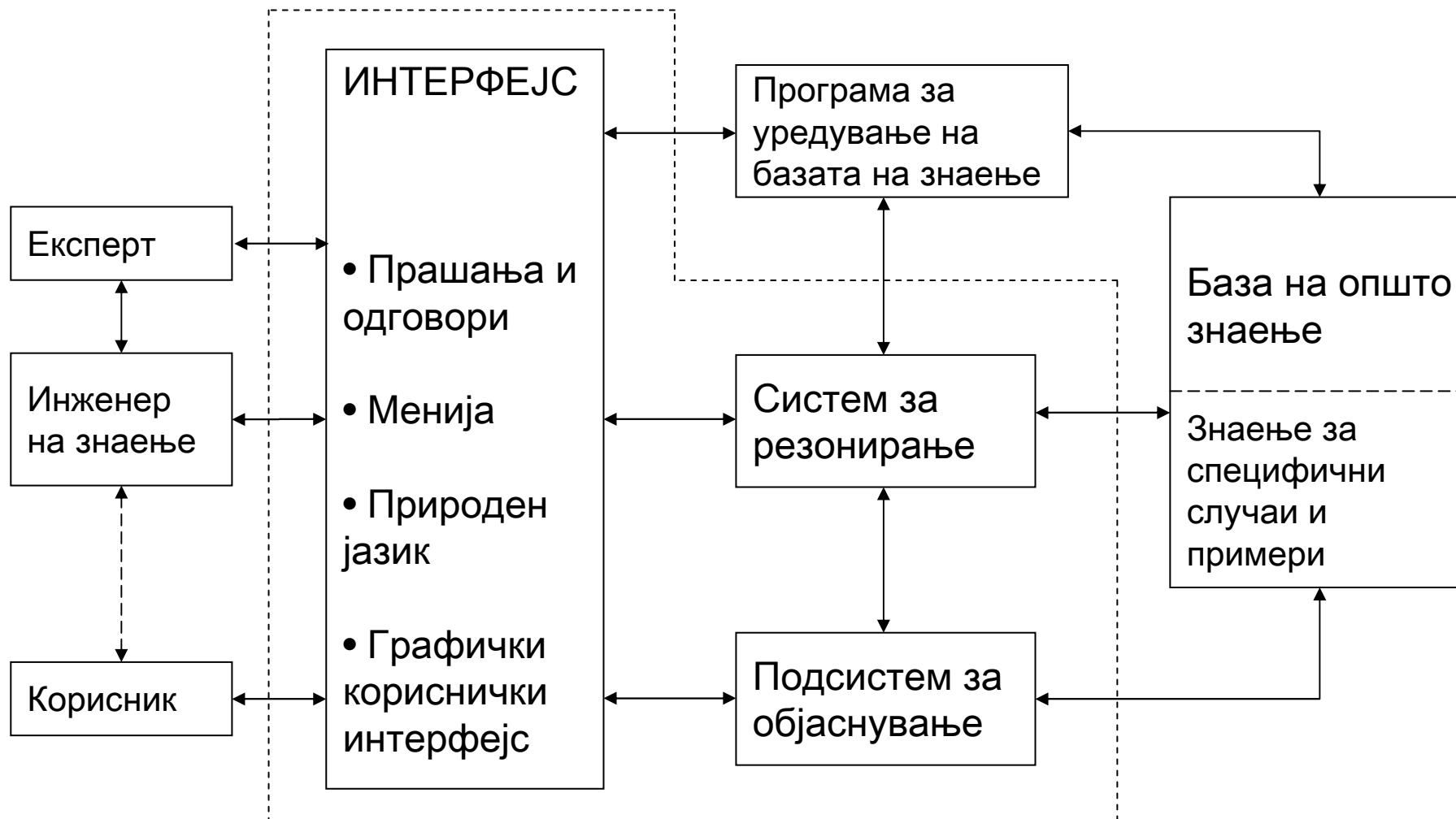
# Шема на структурата на ЕС според Нилсон



# Дизајн на ЕС базиран на правила (на англиски)



# Дизајн на ЕС базиран на правила (на македонски)



# Структура на експертните системи:

- база на знаење составена од
  - база на правила за заклучување и
  - база на факти
- машина за донесување заклучоци
- контролна програма
- модул за усвојување на знаењето
- модул за објаснување



# Јадро на експертниот систем

- Јадро на експертниот систем се:
  - ☐ базата на знаење (долготрајна меморија)
  - ☐ базата на податоците (табла или краткотрајна меморија) и
  - ☐ машината за заклучување.

# База на знаењето

- Базата на знаењето го содржи општото знаење од областа на проблемот.
- Најчесто применуван формализам се производните правила, т.е. врската меѓу причините со последиците, премисите со заклучоците или условите со акциите.
- Овие правила најчесто се од обликот “Ако А, тогаш Б”, при што А и Б можат да бидат и класичните множества, но и неодредените множества.



# База на метазнаењето

- Производните правила во кои е вклучена неодреденоста можат да се сместат во дополнителен сегмент од експертниот систем, наречен база на метазнаењето. Овој дел содржи правила за тоа како да се користат правилата од базата на знаењето.
- Примарната цел на метаправилата е да го упрости решавањето по пат на разделување на непотребните патишта од просторот на пребарување.

# Интерактивно заклучување

- Машината за заклучување кај експертниот систем работи над низа од производни правила и врши заклучување.
- Во модулот за усвојување на знаењето, кај некои експертни системи, дозволено е по пат на интеракција со корисникот да се обновуваат базите на знаење или метазнаење.
- Во принцип, во овој дел мора да се применат соодветни алгоритми на машинското учење.

# Апроксимативно расудување

- Апроксимативното расудување се темели на fuzzy импликацијата, чија форма е:

$$f : [0, 1] * [0, 1] \rightarrow [0, 1]$$

- Импликацијата се дефинира на многу различни начини, како на пример:

- $f(a,b) = -a \vee b$   $a, b \in [0,1]$
- $f(a,b) = \max\{x \in [0,1] \mid a \wedge x \leq b\}, a, b \in [0,1]$
- $f(a,b) = -a \vee (a \wedge b)$
- $f(a,b) = (-a \wedge -b) \vee b$

# Школка на експертниот систем

- Кога од експертниот систем ќе се отстрани знаењето, се добива таканаречена школка на експертниот систем (shell).
- Ако знаењето се базира на производни правила што се неодредени, тогаш експертниот систем се смета за fuzzy систем, а расудувањето се вика апроксимативно расудување.

# Фазите на еден експертен систем

- Усвојување на знаењето
- Осветлување на знаењето
- Анализа на усвоеното знаење
- Интерпретација на изолираниот материјал
- Расудување и заклучување

# Усвојување на знаењето

- Усвојување на знаењето е процес на:
  - ☐ осветлување
  - ☐ анализа и интерпретација на знаењето
  - ☐ преобразба на знаењето во соодветно машинско претставување
- Процесот на усвојување на знаењето е во постојана повратна спрега со експертот (експертите) кои учествувале во неговото креирање.

# Методи за осветлување на знаењето

- Интервју
- Гласно размислување (вербални протоколи)
- Интроспективни протоколи
- Студии на набљудување

# Интервју

- неструктурирани - фокусирани
  - осветлување во глобални рамки
- структурирани
  - суштински концепти и модели на знаењето
- Предности:
  - релаксираност, неформалност
  - усвојување површинско и експлицитно знаење
- Недостатоци:
  - невозможност да се откријат важни детали
  - субјективност на експертот
  - прикривање професионални тајни



# Гласно размислување (вербален протокол)

- Презентација од страна на експертот
- Заедничка анализа со инженерот
- Предности:
  - одредува кога и како да се пристапи во специфични ситуации
  - екстракција на стратегиите за расудување и поделба на потпроблеми
- Недостатоци:
  - долготрајност при интерпретацијата на долгите низи вербални изјави
  - лоша интерпретација на искуството

# Интроспективни протоколи

- Гласно размислување за хипотетични проблеми
- Се сугерира множество потенцијални решенија
- Предности:
  - скицирање на стратегии и правци
  - елиминација на контрадикторности

# Студии на набљудување

- Следење на работата на експертот во неговата реална работна средина

# Анализа на усвоеното знаење

- Екстракција, групирање на клучните податоци
- Одделување релевантни делови од протоколите
- Дефинирање релации меѓу сегментите
- Формирање текстуален модел на знаењето

# Интерпретација на изолираниот материјал (1)

## ■ Тројки

- ОАВ (Објект, Атрибут, Вредност) со одреден степен на сигурност, наречен фактор на сигурноста
- Значење: атрибутот А на објектот О прима вредност В со ниво на сигурност ФС

## ■ Рамки

- Групирање факти и знаења во засебни модули
- Збирка на знаење што се однесува на даден објект, ситуација или концепт
- Претставени се:
  - графички, со мрежи
  - со име и листа парови (атрибут, вредност)

# Интерпретација на изолираниот материјал (2)

- Семантички мрежи
  - јазли: симболи за објекти, ситуации или концепти
  - лаци: односи меѓу јазлите
- Логика
  - Исказно сметање
  - Предикатно сметање
  - Предикатни функции:
    - исто
    - определено
    - конечно
    - познато
    - присутно

# Интерпретација на изолираниот материјал (3)

- Производни правила
  - најчесто применуван формален систем за претставување на знаењето
  - мали, разбирливи фрагменти
  - модуларност
  - причинско - последични односи
  - применливи при хеуристичното расудување
  - структура: услов и акција

# Начини на расудување (резонирање) кај ЕС

- Различни проблеми бараат различен начин на расудување (резонирање) при нивното решавање:
  - Пребарување водено од целта (goal-driven search)
  - Пребарување водено од податоците (data-driven search)
- Изборот на школката на ЕС зависи од проблемот кој треба да се решава
- Прибирањето и организацијата на знаењето во базата на знаење е посебен проблем



# Нагорна абдукција (Bottom-Up abduction)

- Кога системот треба да донесе дијагноза, тогаш тој се обидува да најде правило од типот:  $disease \Rightarrow symptom$  и пребарува за да го најде симптомот.
- За ограничени домени ова е сосема соодветно, но донесува неочекувани заклучоци кога се применува за пошироки намени.
- На пример, ако пациентот има инфективно заболување, системот може да извлече правила како:  
 $(\leftarrow (heart-attack\ ?x) (funny-electrocardiogram-signals\ ?x))$

# Типичните дијалози на Mycin

---

(1) Patient's name: (first-last)

\*\* FRED SMITH

(2) Sex:

\*\* MALE

(3) Age:

\*\* 55

(4) Have you been able to obtain positive cultures from a site at which Fred Smith has an infection?

\*\*YES

-----infection-1-----

(5) What is the infection?

\*\*PRIMARY-BACTEREMIA

The most recent positive culture associated with the primary-bacteremia (infection-1) will be referred to as:

-----culture-1-----

(6) From what site was the specimen for culture-1 taken?

\*\* BLOOD

The first significant organism from this blood culture (culture-1) will be referred to as:

-----organism-1-----

(7) Enter the identity of organism-1

\*\*\* UNKNOWN

(8) Is organism-1 rod or coccus (etc.)

\*\* ROD

(9) The gram stain of organism-1

\*\* GRAMNEG

...

---

# Пример на консултациски режим ...

ЕС: Како се вика пациентот?

Корисникот: Џорџ Вашингтон.

ЕС: Дали е машко или женско?

Корисникот: машко.

ЕС: Кои се симптомите?

Корисникот: главоболка, треска, висока температура

ЕС: Дали има проблеми со stomакот?

Корисникот: да.

ЕС: Кога е роден болниот?

Корисникот: 22. 2. 1732 г.

## ... Пример на консултациски режим

ЕС: Најверојатно Џорџ Вашингтон има вирусна инфекција. Препорачувам: два аспирина дневно, пиење многу течност, многу одмор. Јавете ми за развојот на состојбата на болниот по два дена.

??? Недостиг на општо познати знаења



# Подсистем за објаснување

- Подсистемот за објаснување мора да може да одговара на два вида прашања:
- HOW queries – објаснување на начинот како се добиени заклучоците на системот
- WHY queries – објаснување за тоа зошто на системот му треба одреден податок (знаење)

# Пример на објаснувачки режим...

ЕС: Дали менингитисот е добиен за време на престој во болницата?

Корисникот: Зошто го прашуваш тоа?

< ЕС влегува во објаснувачки режим >

[2.0] Тоа ќе помогне во одредување на организмите што ја предизвикаа инфекцијата. Досега е востановено:

[2.1] инфекцијата е менингитис

## ...Пример на објаснувачки режим...

[2.2] не се забележани организми при првиот наод

[2.3] типот на инфекцијата е бактериска

[2.4] неврохирургијата не е еден од релевантните настани во историјата на пациентот ПЦ100

[2.5] староста на пациентот ПЦ100 е поголема од 25 години

## ... Пример на објаснувачки режим...

Значи ако инфекцијата е настаната додека пациентот ПЦ100 е хоспитализиран, тогаш постои доказ дека организмите кои можеби ја предизвикале се *E. Коли* (0,75), стафилококи (0,3), псевдомона (0,3), клебсиела (0,5).



## ...Пример на објаснувачки режим

Корисникот: Како го доби [2.3]?

ЕС: Следните правила придонесоа кон заклучокот дека инфекцијата е бактериска:

[2.7] правило 148 со доверливост (0,19)

[2.8] правило 500 со доверливост (0,51)

[2.9] правило 501 со доверливост (0,95)

[3.0] правило 502 со доверливост (0,97)

# Што значи ова?

## ■ Користено правило:

□ If:

1. The gram stain of the organism is negative
2. The morphology (shape) is rod
3. The aerobicity is anaerobic

□ Then: The organism is bacteroides.

## ■ Приказ со логичкото правило:

( ← (is ?x bacteroides)

(and (morphology ?x rod) (gram-stain ?x neg)  
(aerobicity ?x anaerobic)))

# Како се добива одговорот ?

- Се бара доказ на целта  
(Show: (is organism-1 ?what))
- Одговорот може да го даде лекарот, но може и да изостане.
- Во тој случај, се преминува кон натамошно генерирање прашања.

# LI фактор

- Кон правилата може да се додаде таканаречениот LI фактор, дефиниран со  $LI = \log(I(d/si))$ , каде  $I(d/si) = P(si/d)/P(si)$ .
- Освен логаритамскиот фактор, вообичаено е да се користи и *факторот на сигурноста* (certainty factor).
- Разликите донесени врз основа на веројатностите, логаритамските и сигурносните фактори се незначителни, така што може да се смета дека Mycin има Bayes-овски карактер.

# MYCIN (~1970)

- Напишан во дијалект на LISP

- Факти од облик:

  - (identity organism\_1 klebsiella 0.25)

  - (sensitivity organism\_2 penicillin -1.0)

- Правила од облик:

  - IF: (AND (same\_context infection primary\_bacteria)  
(membf\_context site sterilesite)  
(same\_context portal GI))

  - THEN: (conclude context\_ident bacteroid tally 0.7)

# Правилата во Mycin

---

```
( ← (is ?x streptococcus-group-a) LI=3
      (and (sterile (site ?x))
            (is ?x streptococcus)
            (portal ?x throat)))
```

```
( ← (is ?x neisseria) LI=5
      (and (gram-stain ?x neg)
            (morpholgy ?x coccus)))
```

```
( ← (contaminant ?x) LI=5
      (and (sterile (site ?x))
            (or (is ?x bacillus-subtilis)
                 (is ?x diphtheriae))))
```

---

# Правила и расудување (резонирање) кај MYCIN

- Без веројатносните коефициенти правилата би биле еквивалентни на предикатна логика
- MYCIN е воден од целта (goal-driven), односно поврзува наназад – тргнува од некоја претпоставка дека инфекцијата е предизвикана од некоја бактерија и се обидува да го докаже тоа

# Несигурностите во системот...

- Фактор на сигурност (доверливост)  
(fidelity, reliability, confidence)
- Опсегот е: од -1            сигурно неточно,  
                  до 1            сигурно точно

Изведување:

IF A THEN B ( $f_{AB}$ )

A ( $f_A$ )  $\Rightarrow$  B ( $f_B = f_A \cdot f_{AB}$ )

Или со прагова вредност

(ако  $f_A > \Theta$  тогаш  $f_B = f_A \cdot f_{AB}$ )    ( $\Theta \approx 0,2$ )



# Предности

- Нуди постојано ист одговор за повторливи одлуки, процеси и задачи
- Поседува и одржува значајно количество информации
- Ја поттикнува организацијата за да ја прикаже логиката според која ги донесува одлуките.
- Никогаш не заборава да постави прашање, што кај човечкиот експерт може да изостане.

# Недостатоци

- Му недостасува општо знаење (common sense) за да донесе одлука.
- Не може да даде толку креативни одговори какви што би дал човекот, посебно во непредвидени ситуации.
- Експертите во доменот понекогаш не се во состојба да ја објаснат својата логика и начинот на кој расудуваат.
- Во дефинирањето на базата знаење може да се појават грешки кои водат кон нелогични одлуки.
- Сосема е неприлагодлив за променлива околина, освен ако не научат како да ја променат базата на знаењето.

# Користена литература

- Artificial Intelligence, A Modern Approach  
2nd edition, Russel and Norvig
- Artificial Intelligence, A New Synthesis, Nils J.  
Nilsson

# Прашања?

ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Комуникацијата меѓу агентите

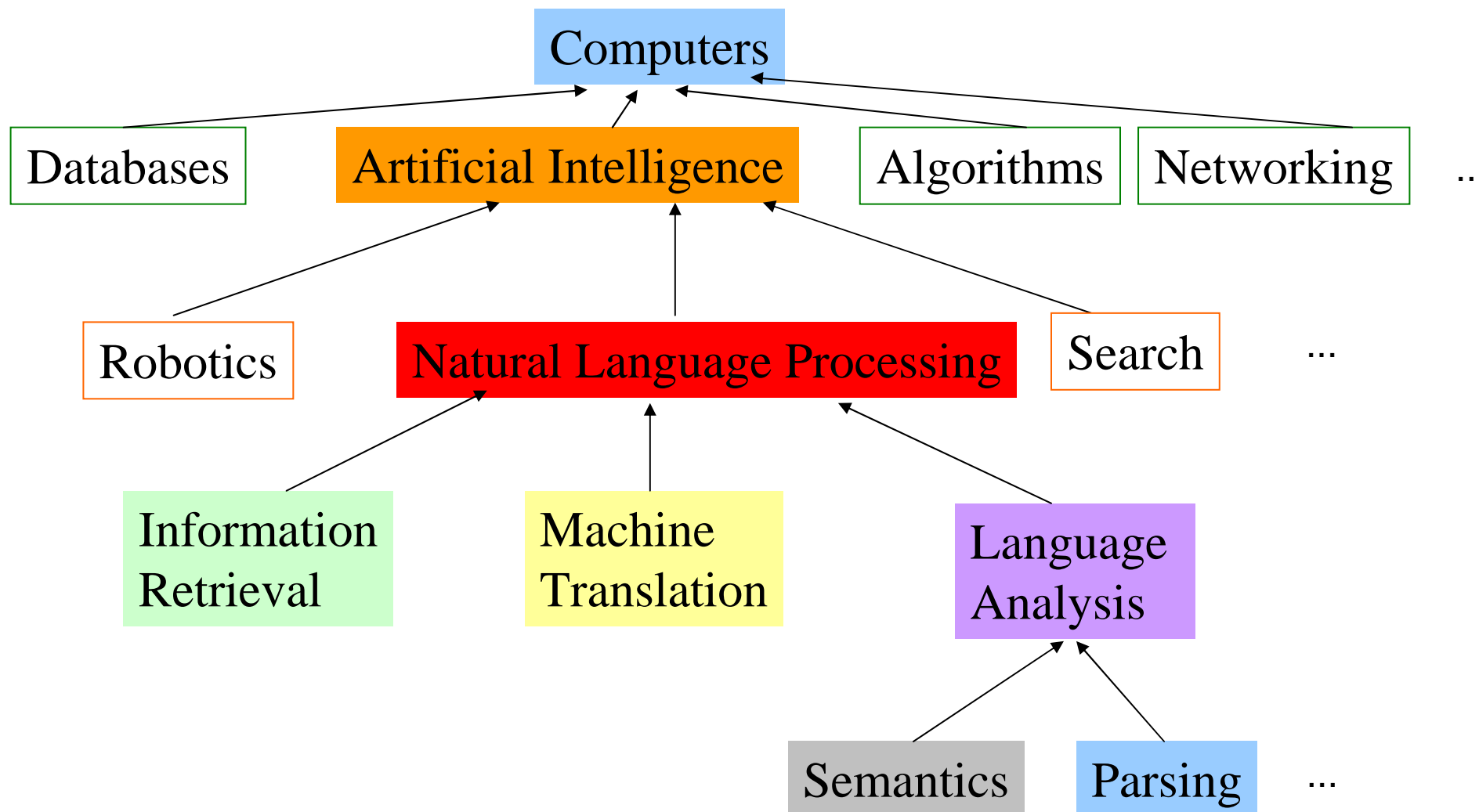
# Комуникација

- Дефиницијата од Википедија:  
Комуникацијата претставува процес на пренос на информациите од испраќачот кон примателот со помош на таков медиум преку кој информацијата која се препраќа ја разбираат и испраќачот и примателот.
- Притоа, ако комуникацијата се одвива меѓу луѓето, тогаш испраќачот е говорник, медиумот се акустичните бранови, а примателот е слушателот.

# Обработка на природни јазици (Natural Language Processing)

- Првенствено се проучуваат сите нешта што се неопходни за да се овозможи заемодејство (интеракција) со компјутерите на многу поприроден начин (со користење на природен јазик)
- Но, исто така, ваквите компјутерски обработки на природните јазици, даваат идеи за можните начини како луѓето ги обработуваат јазиците

# Каде се вклопува во поделбата на компјутерските науки?





# Актуелни примени на обработката на природни јазици

- Одговарање на прашања  
(Question answering)
- Конверзациски агенти  
(Conversational agents)
- Прераскажување (резимирање)  
(Summarization)
- Анализа на оговарања и расположение  
(Buzz analysis, Sentiment analysis)
- Машинско преведување  
(Machine translation)

# Трите основни семиотски правила

- Семиотиката е наука која се занимава со проучување на знаковните процеси и со тоа како се создава и сфаќа значењето.
- Трите основни правила на семиотиката се:
  1. синтаксата
  2. прагматиката и
  3. семантиката

# Синтакса

- Синтаксата е наука која се занимава со наоѓањето на правилата според кои се конструираат речениците во еден јазик.
- На пример, правилото според кое е конструирана реченицата “Времето е многу топло.” е: именка, глагол, прилог, придавка.
- А правилото на реченицата “Топлото време им пречи на луѓето.” е: придавка, именка, кратка заменска форма, глагол, предлог, именка.

# Прагматика

- Прагматиката е филозофски правец кој се занимава со тоа на каков начин се користат зборовите при комуникацијата.
- На пример, реченицата “Моливот ја преврте кравата” е граматички сосема исправна, но не носи порака, бидејќи навидум нема смисол. Но, ако на оваа реченица и претходи реченицата “Весна го фрли моливот кон гумената крава.”, тогаш реченицата станува прифатлива.

# Семантика

- Семантиката е наука која се занимава со значењето на зборовите (лексичка семантика), комбинациите зборови (фразеологија) и како фразите се комбинираат за да може да се одреди значењето на реченицата.
- На пример, дали го знаете значењето на реченица која започнува со: “Петар јаде буреќ со ...”, а на местото на празните места има само една именка?

# Петар јаде бурека со ...

- месо.
- прсти.
- Марко.
- јогурт.
- неконгруентен атрибут
- инструмент
- објект (субјект)
- објект (објект)
- Секоја од четирите реченици има еднаква синтакса, но семантиката им е сосема различна.

# Претставување на синтаксата

## BNF = Backus-Naur form

- Реченицата се разбива на помали целини (нетерминални симболи).
- Тие постепено се трансформираат кон поелементарни нетерминални симболи кои ги претставуваат зборовните групи (PoS = Part of Speech).
- На крајот, овие нетерминални симболи се трансформираат во терминални симболи кои ги претставуваат зборовите во реченицата.

# Силното сонце го стопи снегот.

Подготовка за BNF:

- $S = \text{подмет (NP)} + \{\text{прирок} + \text{предмет}\} (VP)$
- NP = силното сонце
- VP = го стопи снегот
- силното – придавка Adj
- сонце – именка N
- го – кратка заменска форма Pron
- стопи – глагол V
- снегот – именка N

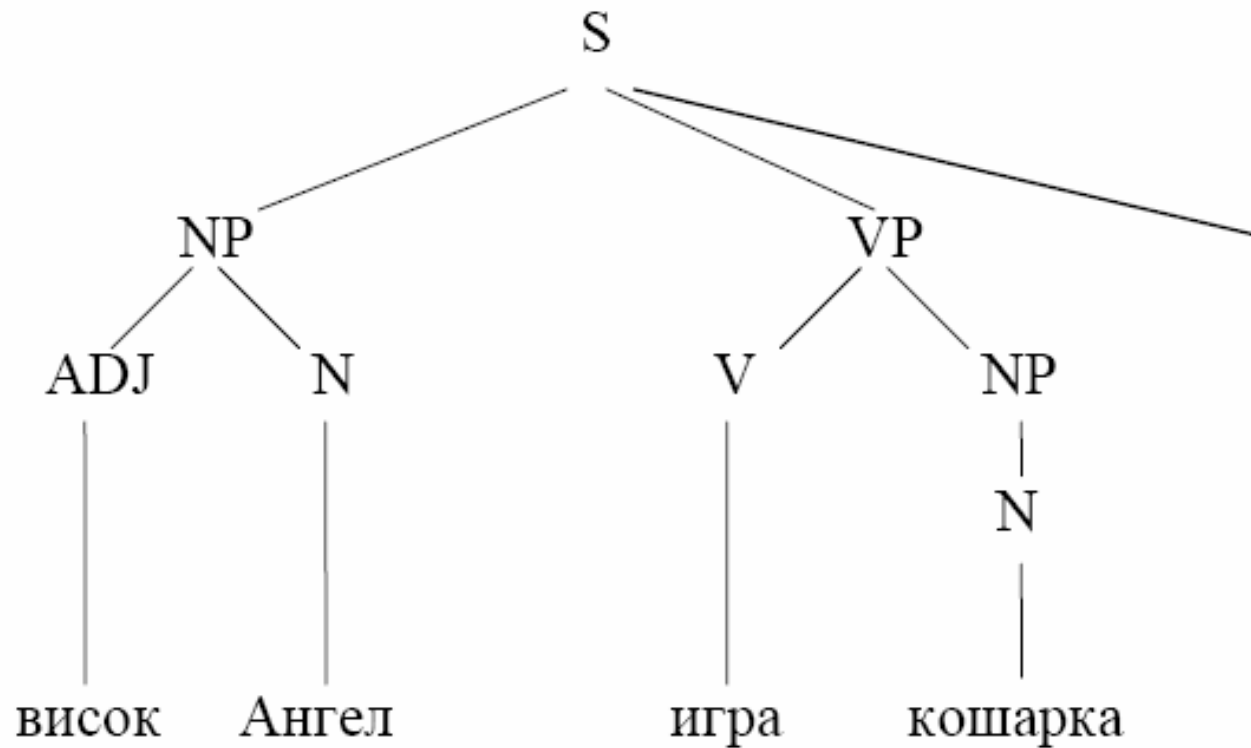


# Високиот Ангел игра кошарка.

Подготовка за BNF:

- $S = \text{подмет (NP)} + \{\text{прирок} + \text{предмет}\} (VP)$
- $NP = \text{високиот Ангел}$
- $VP = \text{игра кошарка}$
- високиот – придавка Adj
- Ангел – именка N
- игра – глагол V
- кошарка – именка N

# Графички приказ: длабока структура (deep structure)



# Некои поважни ознаки во BNF

- NP – noun phrase (именска фраза)
- VP – verb phrase (глаголска фраза)
- N – noun (именка)
- V – verb (глагол)
- Adj – adjective (придавка)
- Adv – adverb (прилог)
- Prep – Preposition (предлог)
- Conj – conjunction (сврзник)
- PP – personal pronoun (лична заменка)
- Det – determiner (член)

# Контекстно-слободни граматики

- **N** множество нетерминални симболи (или променливи)
- $\Sigma$  множество терминални симболи (различно од N)
- **R** множество од продукции или правила од обликот  $A \rightarrow \alpha$ , каде  $A$  е нетерминален симбол, а  $\alpha$  е стринг од симболи од  $(\Sigma \cup N)^*$
- **S** предодреден нетерминален симбол наречен почетен симбол

# Контекстно-слободни граматики

## Граматика

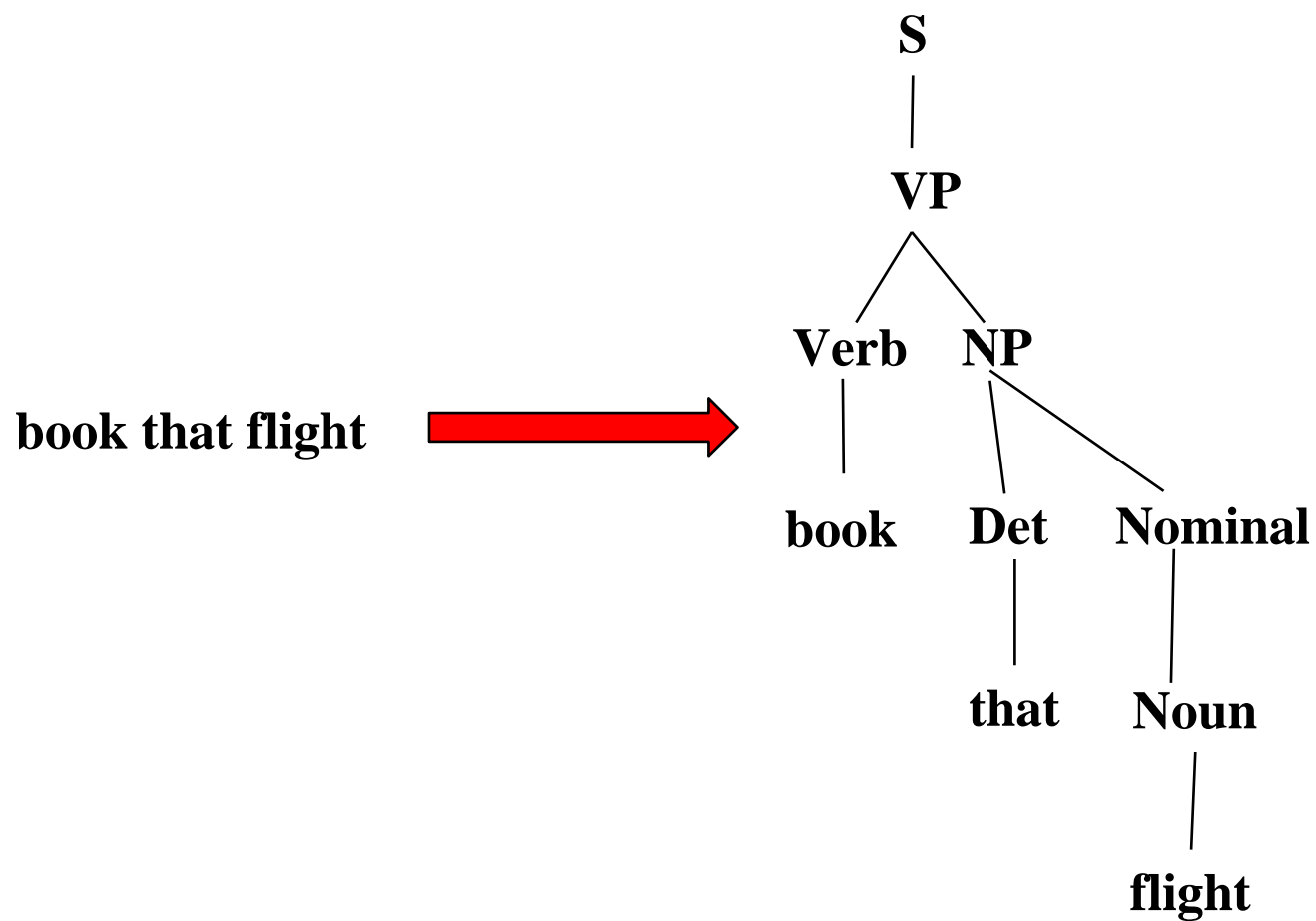
**S** → **NP VP**  
**S** → **Aux NP VP**  
**S** → **VP**  
**NP** → **Pronoun**  
**NP** → **Proper-Noun**  
**NP** → **Det Nominal**  
**Nominal** → **Noun**  
**Nominal** → **Nominal Noun**  
**Nominal** → **Nominal PP**  
**VP** → **Verb**  
**VP** → **Verb NP**  
**VP** → **VP PP**  
**PP** → **Prep NP**

## Лексикон

**Det** → **the | a | that | this**  
**Noun** → **book | flight | meal | money**  
**Verb** → **book | include | prefer**  
**Pronoun** → **I | he | she | me**  
**Proper-Noun** → **Houston | NWA**  
**Aux** → **does**  
**Prep** → **from | to | on | near | through**

Пример граматика и лексикон  
за корпусот ATIS за контрола  
на авионски летови

# Пример



# Парсирање

- Одредувањето на синтаксичката исправност на реченицата се вика парсирање.
- Тоа значи дека се одредува дали редоследот на зборовните групи е точен или не.
- Парсирањето може да биде надолно (Top down) или нагорно (Bottom up)
- При надолното парсирање се тргнува од  $S$  кон збороформите
- При нагорното парсирање се тргнува од збороформите кон  $S$

# Пример

Дадена е следнава регуларна граматика:

$\langle S \rangle ::= \langle NP \rangle \langle VP \rangle$

$\langle NP \rangle ::= \langle ADJ \rangle \langle NP \rangle \mid \langle N \rangle$

$\langle VP \rangle ::= \langle ADV \rangle \langle VP \rangle \mid \langle V \rangle \langle NP \rangle \mid \langle V \rangle$

$\langle N \rangle ::= \text{Mare} \mid \text{Kiro} \mid \text{Toni} \mid \text{pesna} \mid \text{slatko}$

$\langle V \rangle ::= \text{pee} \mid \text{jade}$

$\langle ADJ \rangle ::= \text{ubavo} \mid \text{golemo} \mid \text{slatko} \mid \text{vkusno}$

$\langle ADV \rangle ::= \text{ubavo} \mid \text{vkusno} \mid \text{brzo}$



Дали со неа се парсира реченицата  
“Маре јаде убаво вкусно слатко слатко”?

<S> => <NP> <VP>

=> <N> <VP>

=> Mare <VP>

=> Mare <V> <NP>

=> Mare jade <NP>

=> Mare jade <ADJ> <NP>

=> Mare jade ubavo <NP>

=> Mare jade ubavo <ADJ> <NP>

=> Mare jade ubavo вкусно <ADJ> <NP>

=> Mare jade ubavo вкусно слатко <NP>

=> Mare jade ubavo вкусно слатко <N>

=> Mare jade ubavo вкусно слатко слатко

# Нагорно парсирање на реченицата “Маре јаде убаво вкусно слатко слатко”?

Mare  $\leq$   $\langle N \rangle$

jade  $\leq$   $\langle V \rangle$

ubavo  $\leq$   $\langle ADV \rangle$  |  $\langle ADJ \rangle$

vkusno  $\leq$   $\langle ADV \rangle$  |  $\langle ADJ \rangle$

slatko  $\leq$   $\langle N \rangle$  |  $\langle ADJ \rangle$

$\langle N \rangle \leq \langle NP \rangle$

оттука:

slatko  $\leq \langle NP \rangle$

$\langle V \rangle \langle ADV \rangle$ ,  $\langle V \rangle \langle ADJ \rangle$

отпаѓаат

$\langle ADV \rangle \langle ADV \rangle$ ,  $\langle ADV \rangle \langle ADJ \rangle$ ,  
 $\langle ADJ \rangle \langle ADV \rangle$  и  $\langle ADJ \rangle \langle ADJ \rangle$   
исто така отпаѓаат

Како пар ADJ и NP го редуцираат  
NP.

$\langle ADJ \rangle \langle NP \rangle \leq \langle NP \rangle$

Досега се извршени следниве  
редукции:

Mare  $\leq \langle NP \rangle$                       jade  $\leq$   
 $\langle V \rangle$

slatko  $\leq \langle NP \rangle$                       slatko

slatko  $\leq \langle NP \rangle$

$\langle ADV \rangle \langle NP \rangle$  отпаѓа

Од тоа што постои редукцијата:

$\langle ADJ \rangle \langle NP \rangle \leq \langle NP \rangle$

се генерира:

ubavo vkusno slatko slatko  $\leq \langle NP \rangle$

Следуваат:

$\langle V \rangle \langle NP \rangle \leq \langle VP \rangle$

$\langle NP \rangle \langle VP \rangle \leq \langle S \rangle$

# Задача:

Користејќи ја истата регуларна граматика, да се изврши нагорно и надолно парсирање на речениците:

1. Тони убаво пее убава песна.
2. Киро убаво слатко јаде големо вкусно Тони.
3. Убавата Маре убаво пее убава песна.
4. Маре јаде убаво.

# Повеќезначност на толкувањето (1)

- Што значи фразата:  
EXTRA WIDE VEHICLE ENTRANCE
- extra (wide (vehicle entrance))  
Дополнителен широк влез за возила
- (extra wide) (vehicle entrance))  
Исклучително широк влез за возила
- extra (wide vehicle) entrance  
Дополнителен влез за широки возила

## Повеќезначност на толкувањето (2)

- Што значи реченицата:

На полицајците им беше наредено да го прекинат пиењето во 12 часот.

- Тие пиеле па им било наредено да престанат да пијат.
- Им било наредено да влезат во крчмата во 12 часот и да направат фајронт.
- Во 12 часот им било наредено да направат фајронт.

# Природните јазици и комуникацијата

- Трите основни правци на обработката на природните јазици (**Natural Language Processing, NLP**) се:
  1. Разбирањето / сфаќањето (**Understanding / Comprehension**)
  2. Генерирањето / создавањето (**Generation / Production**)
  3. Преведувањето (**Translation**)

# Разбирање

- Препознавање на говорот  
(Speech recognition)
- Синтаксичка анализа  
(Syntactic analysis)
- Семантичка анализа  
(Semantic analysis)
- Прагматична анализа  
(Pragmatic analysis)

# Генерирање и преведување

## ■ Генерирање

- ☐ Избор на содржини (Content selection)
- ☐ Синтаксичка реализација (Syntactic realization)
- ☐ Синтеза на говорот (Speech synthesis)

## ■ Преведување

- ☐ Разбирање (Understanding)
- ☐ Генерирање (Generation)



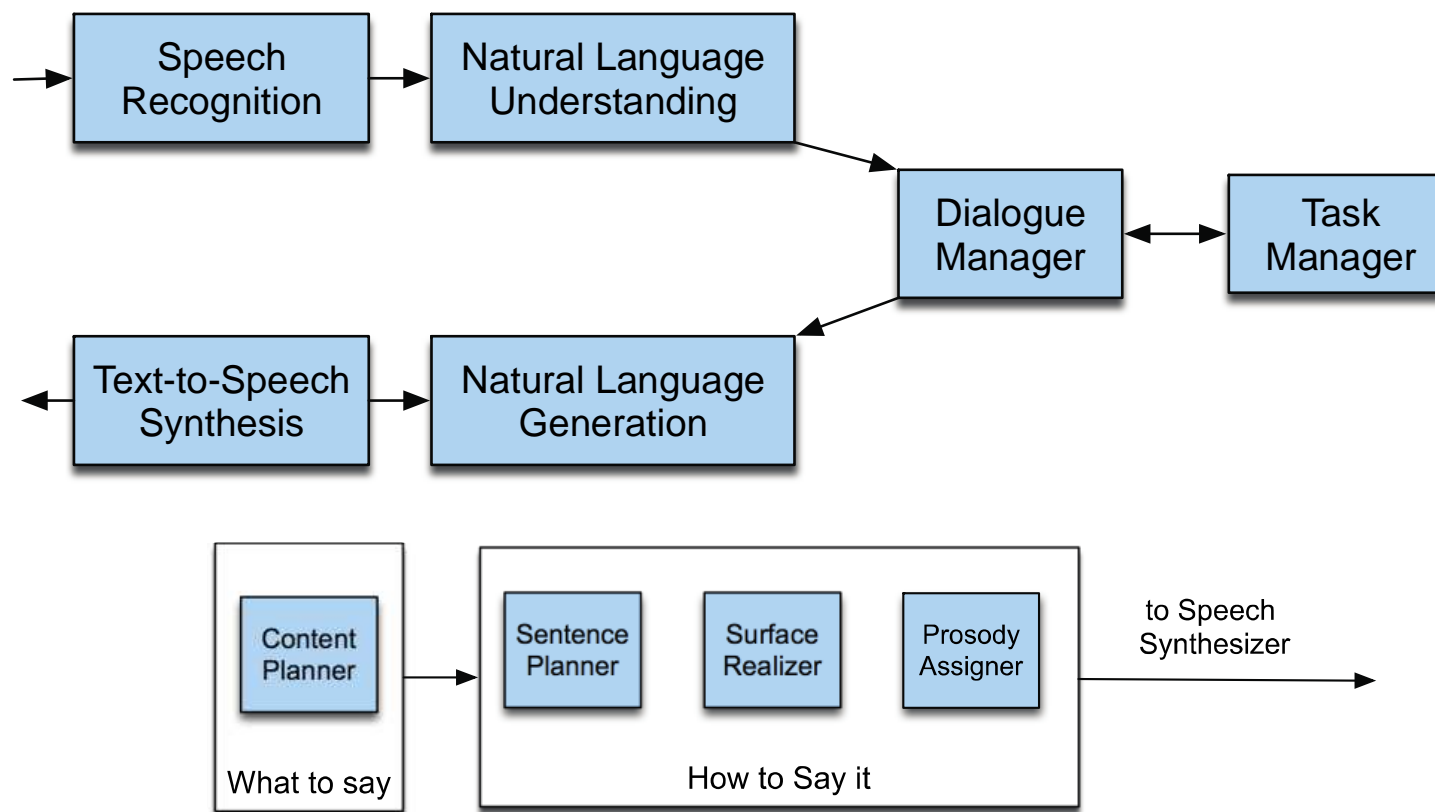
# Разговорни агенти

- Што претставува еден разговорен дијалог систем?
  - Систем кој му овозможува на корисникот да ги изговара неговите прашања на природен јазик и на сличен начин да го добива одговорот од системот.
  - Говорните дијалог системи обезбедуваат интерфејс помеѓу корисникот и компјутерски-базираната апликација која дозволува говорна интеракција со апликацијата на релативно природен начин.



# Разговорни агенти

## ■ Архитектура на еден разговорен агент



# Повеќезначност и хомонимија

- Два збора со иста форма и различни значења се повеќезначни.

Пример: Петре има убава коса.

коса (алатка)

коса (дел од главата)

- Две или повеќе збороформи со иста форма се хомонимни

Пример: убаво дете убаво пее

убаво (придавка, опис на именката)

убаво (прилог, опис на глаголот)

# Зборовна повеќезначност

- Разрешување на повеќезначноста  
(Word Sense Disambiguation, WSD)
- Најсоодветен пристап: научи го  
системот да врши добра класификација
- Алгоритми за учење:
  - ☐ веројатносно расудување (класичен Bayes)
  - ☐ логика од прв ред
  - ☐ невронски мрежи

# Пристап

- Во речникот на секој јазик се дадени зборовните групи (PoS) на заглавните зборови
- Одредувањето на точната зборовна група во рамките на реченицата е проблем на класификација
- Контекстот во кој се очекува повеќезначноста треба да се претстави со помош на множество својства за нивно разрешување (на пр. редоследот на зборовните групи во реченицата, или претходниот контекст)
- Обучувањето на класификацијата се врши со помош на множество за обука

# Одредување на зборовната група (Part Of Speech Tagging)

- Да се класифицира секоја збороформа независно, користејќи ги информациите од речникот и информациите за збороформите во околината.
- Означувањето повторно претставува класификација, значи, може да се научи
- Класификацијата може да биде од лево надесно (forward) или од десно налево (backward)
- Најуспешни алгоритми се т.н. Веројатносни последователни модели (Probabilistic sequence models)

# Морфологија

- Синтаксата не е можна без морфологијата
- Со морфологијата се истражуваат збороформите и збороформирањето
- Морфологијата и синтаксата ги одредуваат морфосинтаксичките описи (Morpho-Syntactic Description, MSD) на збороформите во јазикот
- Пример за MSD-ата на Multext: највеселава Aqsfs-p

# Зборовните групи во Multext

<http://nl.ijs.si/ME/Vault/V3/msd/html/>

Part-of-Speech	Code	Atts
Noun	N	10
Verb	V	15
Adjective	A	12
Pronoun	P	17
Determiner	D	10
Article	T	6
Adverb	R	6
Adposition	S	4
Conjunction	C	7
Numeral	M	12
Interjection	I	2
Residual	X	0
Abbreviation	Y	5
Particle	Q	3



# MSD-ата за македонските именки

Атрибут	Вредност	Пример	Код
Тип	општа (common)	човек, жена, дете	c
	сопствена (proper)	Иван, Вера, Скопје	p
Род	машки (masculine)	човек, Иван	m
	женски (feminine)	жена, Вера	f
	среден (neuter)	дете, Скопје	n
Број	еднина (singular)	човек, жена, дете	s
	множина (plural)	луѓе, жени, деца	p
	избројана множина (count)	човека	t
Падеж	номинатив (nominative)	човек, Иван, жена, дете	n
	вокатив (vocative)	човеку, жено	v
	j.c.(l.s.) е-вокатив (e-vocative)	човече, Иване	w
	j.c.(l.s.) а-форма (oblique)	човека, Ивана, ешка	o
Одреденост	не (no)	човек, жена, дете	n
	да (yes)	човекот, жената, детето	y
	j.c.(l.s.) блиска (proximal)	човеков, женава, детево	p
	j.c.(l.s.) далечна (distal)	човекон, женана, детено	d

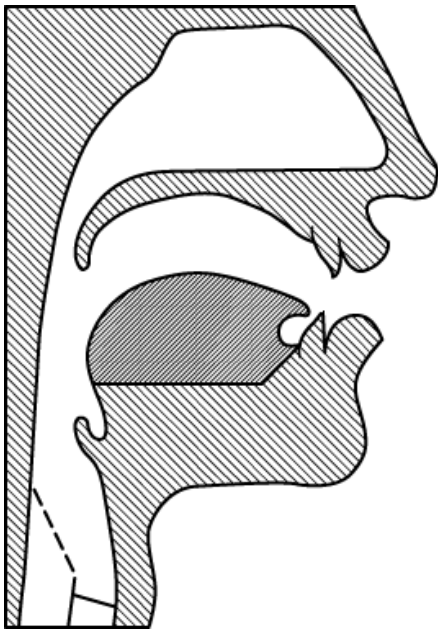
4 Person	first	1	x	x	x	x	x	x	x	x	x	x	x
	second	2	x	x	x	x	x	x	x	x	x	x	x
	third	3	x	x	x	x	x	x	x	x	x	x	x
-----													
5 Number	singular	s	x	x	x	x	x	x	x	x	x	x	x
	plural	p	x	x	x	x	x	x	x	x	x	x	x
	dual	d			x								x
	collective	l											x
-----													
6 Gender	masculine	m		x	x	x	x	x			x	x	x
	feminine	f		x	x	x	x	x			x	x	x
	neuter	n		x	x	x	x	x			x	x	x
* ***** *													
7 Voice	active	a			x	x	x	x	x		x	x	x
	passive	p			x	x	x	x	x		x	x	x
-----													
8 Negative	no	n			x	x		x	x		x	x	x
	yes	y			x	x		x	x		x	x	x
-----													
9 Definiteness	no	n					x				x		
	yes	y					x				x		
	short_art	s					x						
	full_art	f					x						

# WordNet – семантички лексикон

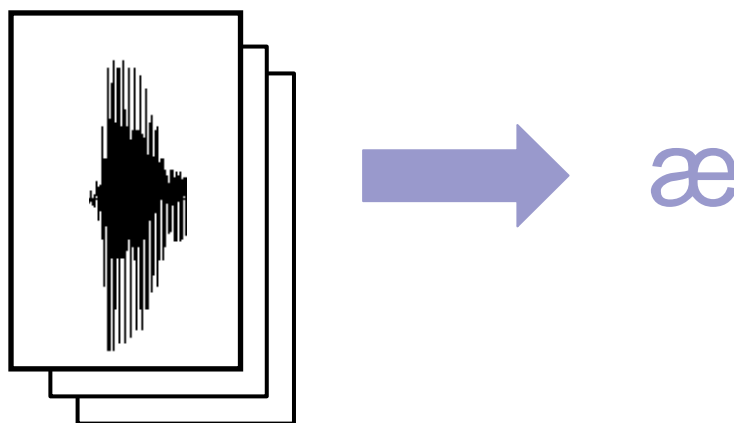
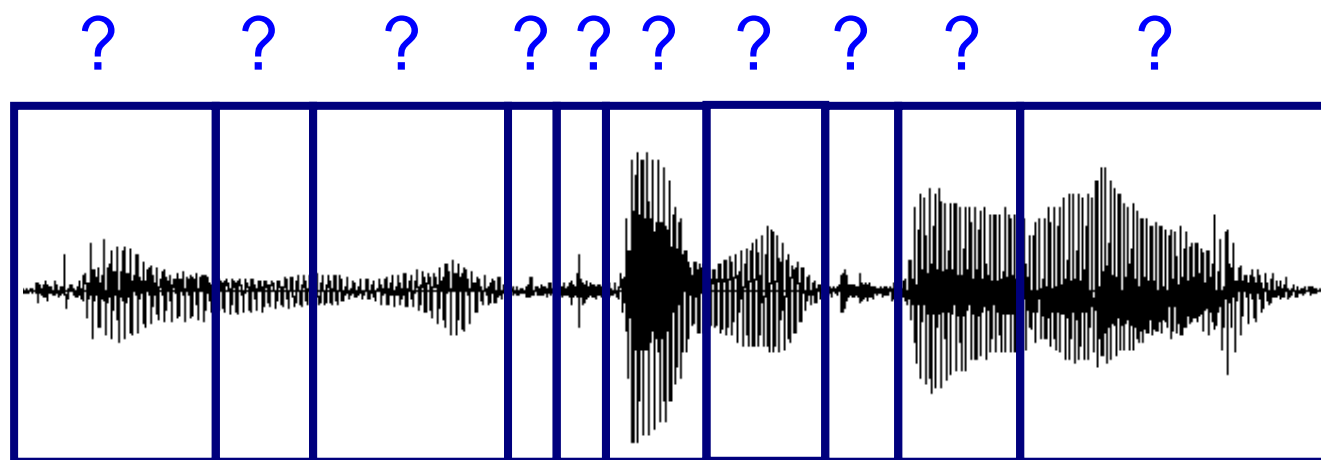
- WordNet е семантички лексикон на јазиците
- Целта му е да направи речник и тезаурус соодветен за интелигентна обработка со помош на компјутер.
- Групирање на зборовите во множества синоними (зборови со исто или слично значење), таканаречени synsets.
- Освен врска меѓу зборовите кои се синоними, за зборовите кои имаат антониими (зборови со спротивно значење) се воспоставуваат и односи кон антониимите.
- Во рамките на син-множествата, меѓу зборовите се дефинираат семантички односи.
- WordNet е организиран хиерархиски.

# Акустично моделирање (Acoustic Modeling)

$$P(words)P(acoustics|words)$$

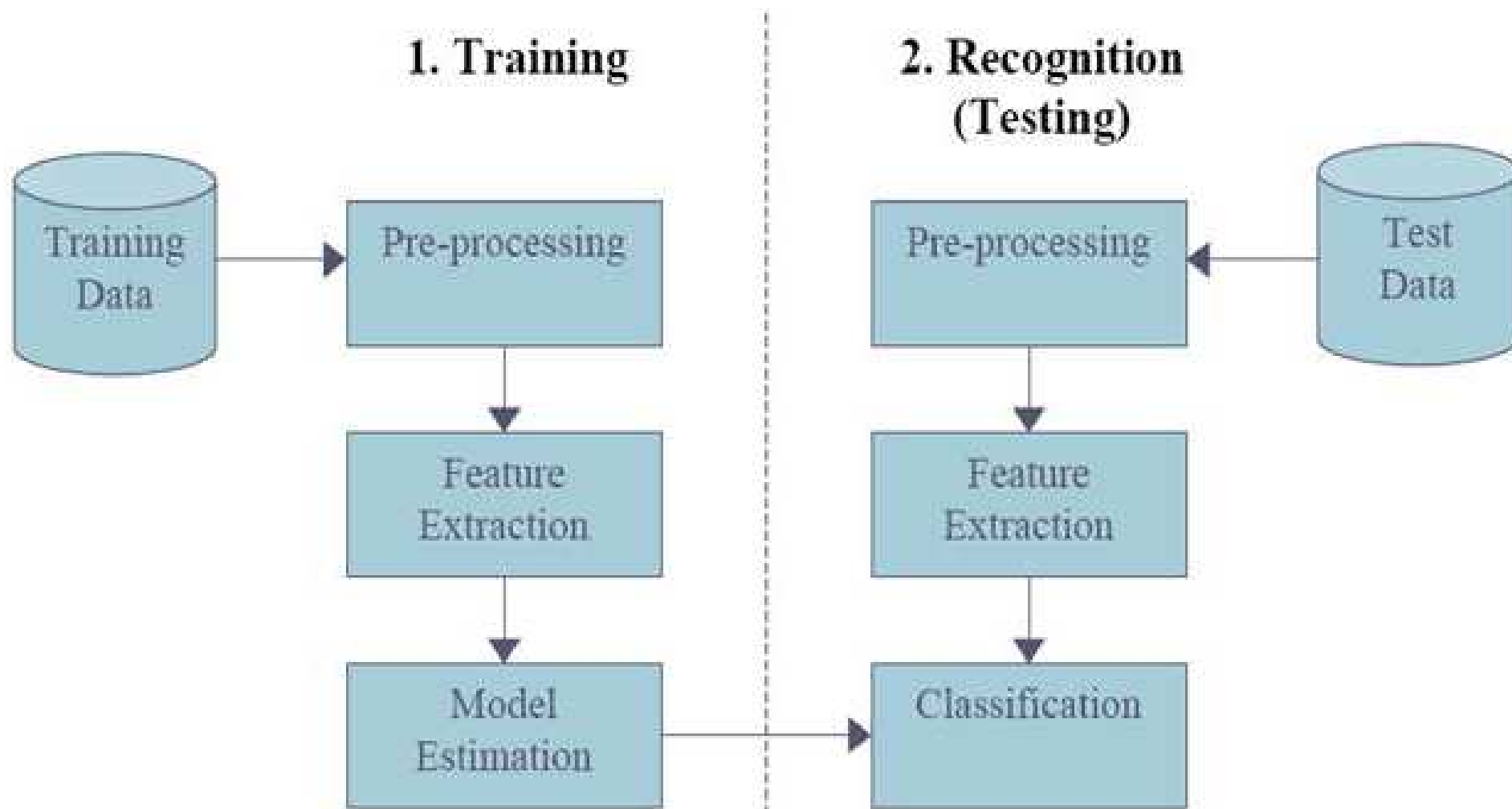


# Класификација на фонеме



Нема јасна граница каде почнува дадена буква (поточно фонема), каде завршува.

# Типичен систем за препознавање на говор



# Делови на системот за препознавање на говор

- Предобработката (pre-processing) обично вклучува откривање на постоење на звук и отстранување на шум
- Извлекувањето на својства (feature extraction) вклучува разни трансформации од временски во фреквентен домен (со варијанти на Фуриеова трансформација), при што сигналот се дели на мали рамки од по 20 ms, поместени за 5 до 10 ms.
- Класификацијата и изградбата на т.н. модел обично се прави со невронски мрежи комбинирани со некаков систем за анализа на секвенци



# Прашања?