



ФАКУЛТЕТ ЗА ИНФОРМАТИЧКИ НАУКИ И КОМПЈУТЕРСКО ИНЖЕНЕРСТВО

# Проблеми кои се сведуваат на пребарување



# Агенти кои решаваат проблеми (според Russel и Norvig)

- Агентите кои се стремат кон тоа да достигнат некоја цел се наречени целно базирани агенти (goal-based agents)
- Нивната цел е од состојбата во која се наоѓаат тековно да ја достигнат целта која е дефинирана како посакувана
- Еден подтип на овие агенти се агентите кои решаваат проблеми (problem-solving agents)
- Нивната цел е да ја најдат најдобрата низа активности која води кон посакуваната цел



# Пример за агент што решава проблеми

```
function SIMPLE-PROBLEM-SOLVING-AGENT(percept) returns an action
    static: seq, an action sequence, initially empty
            state, some description of the current world state
            goal, a goal, initially null
            problem, a problem formulation

    state  $\leftarrow$  UPDATE-STATE(state, percept)
    if seq is empty then do
        goal  $\leftarrow$  FORMULATE-GOAL(state)
        problem  $\leftarrow$  FORMULATE-PROBLEM(state, goal)
        seq  $\leftarrow$  SEARCH(problem)
        action  $\leftarrow$  FIRST(seq)
        seq  $\leftarrow$  REST(seq)
    return action
```

# Што е преbaraување?

- Постапката (или процесот) кој-а овозможува наоѓање на најдобрата низа активности кои водат до одредена цел (според Russel и Norvig)
- Процесот на наоѓање на **барем едно решение** поаѓајќи од почетна (влезна) состојба
- Процесот на наоѓање на **оптималното решение** поаѓајќи од почетна (влезна) состојба



# Што одредува преbaraувањето?

- Кои се решенијата на еден проблем?
- Кое е најдоброто (оптимално) решение на проблемот?
- Дали за проблемот воопшто постои некое решение?



# Проблеми кое се сведуваат на пребарување

- Наоѓање пат меѓу две места
- Наоѓање решение на проблем кој може да се претстави со помош на дрво или со помош на граф
- Играње игри
- Решавање математички проблеми
- Расудување
- Парсирање реченици

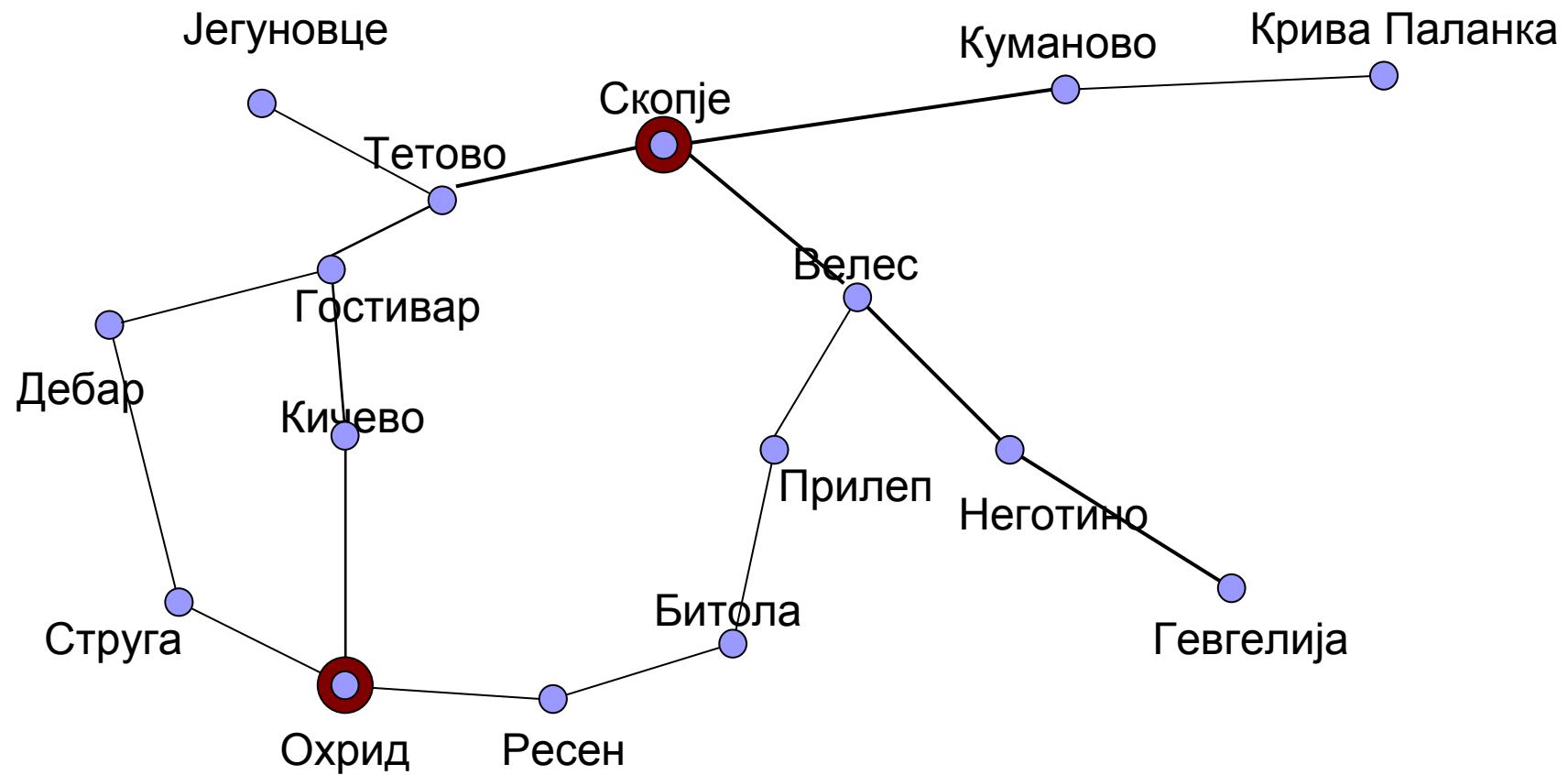


# Како Google Map го наоѓа патот меѓу Скопје и Охрид?



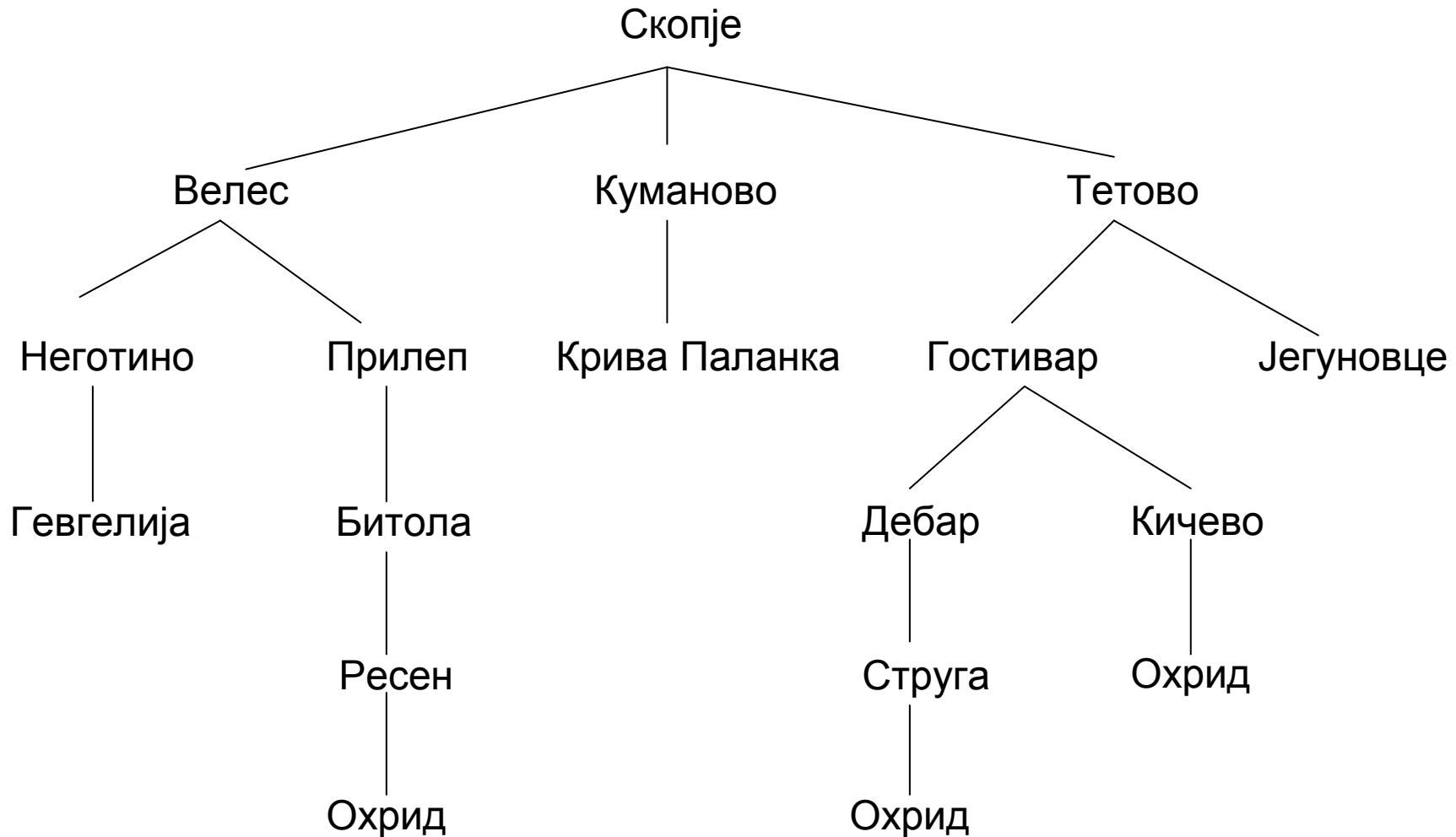


# Наоѓање на патот меѓу Скопје и Охрид





# Дел од дрвото што произлегува од картата



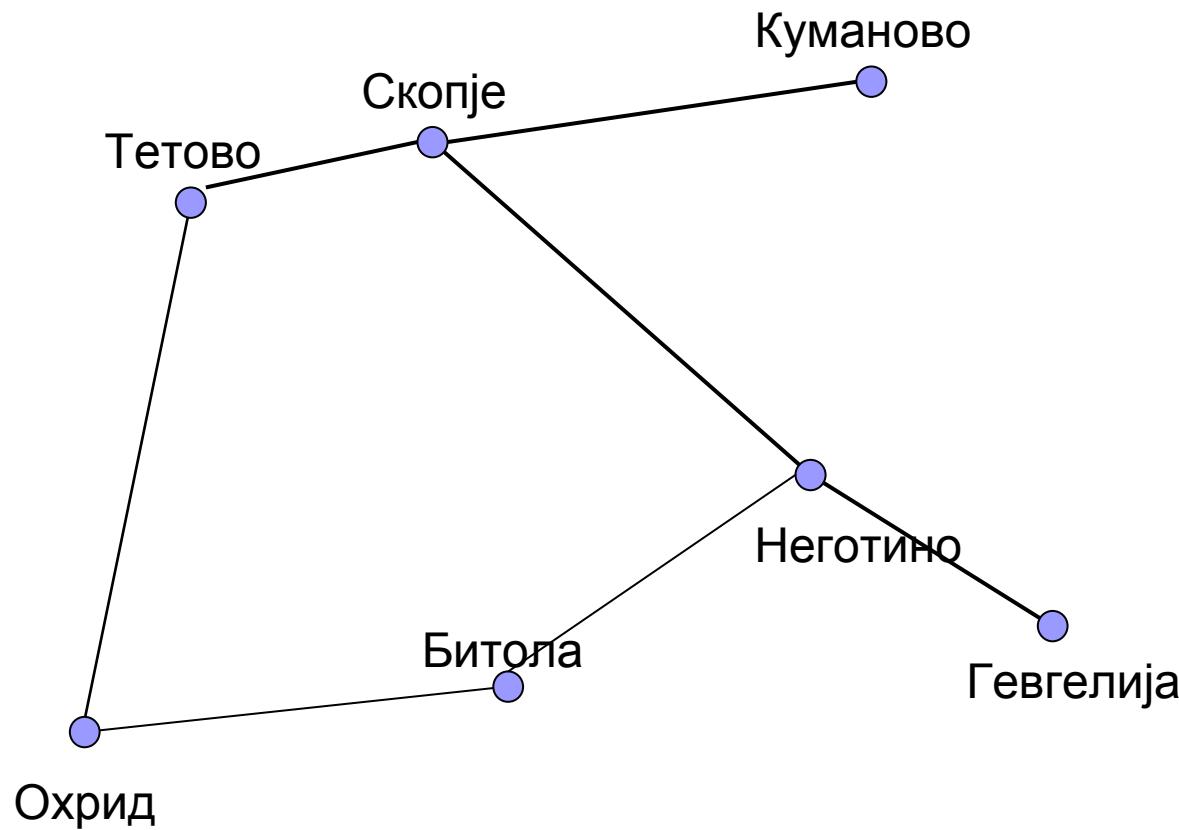


# Критериуми за создавање на дрвото

- Почетна состојба: Скопје
- Крајна цел: Охрид
- Критериум за премин: постои автопат
  
- Не е дефинирана цената на транспортот меѓу две места
- Листата градови со кои има врска еден град се подредени по азбучен ред
- Доколку постои цена, овој критериум може да се промени
- Ограничувачки фактор: нема враќање назад

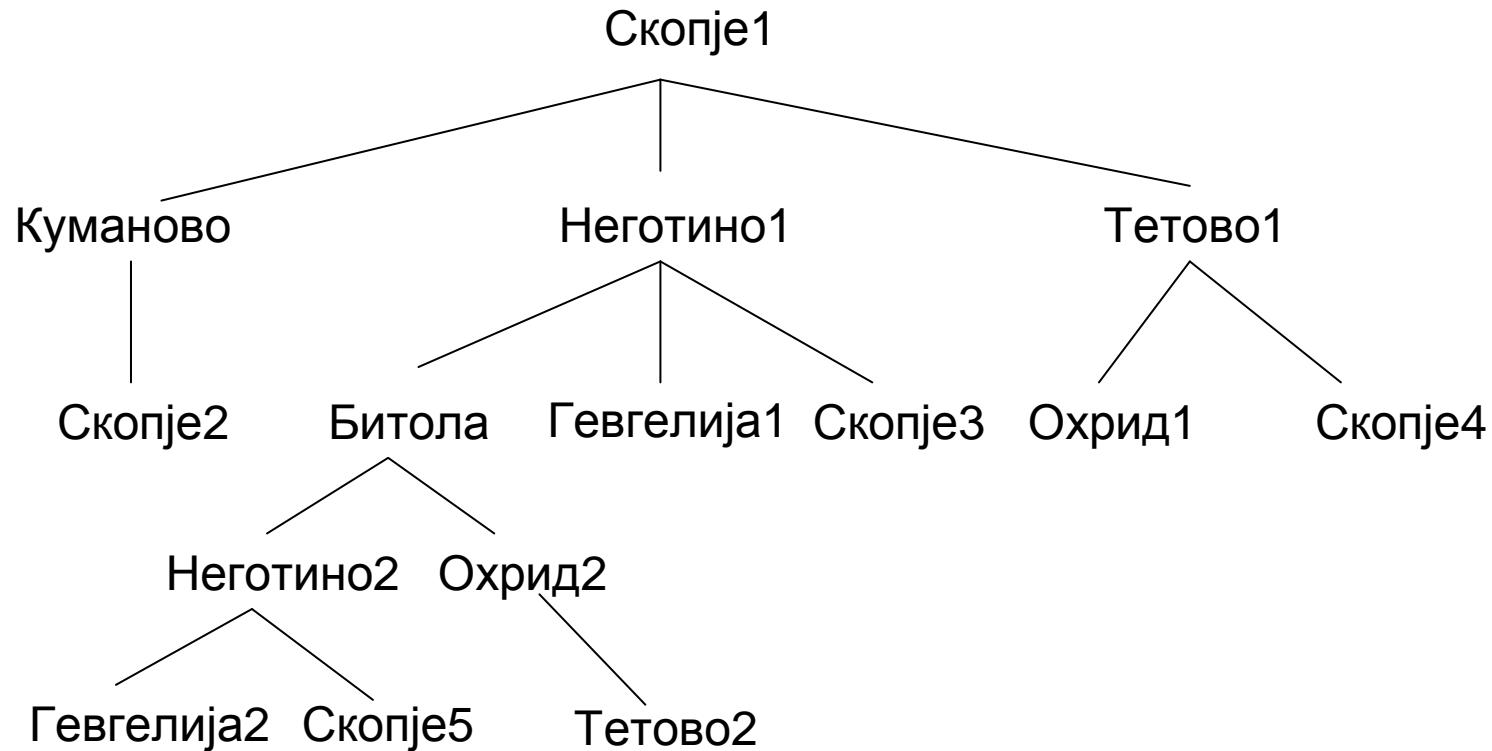


# Наоѓање решение на проблемот на трговскиот патник





# Дел од графот претставен во форма на дрво



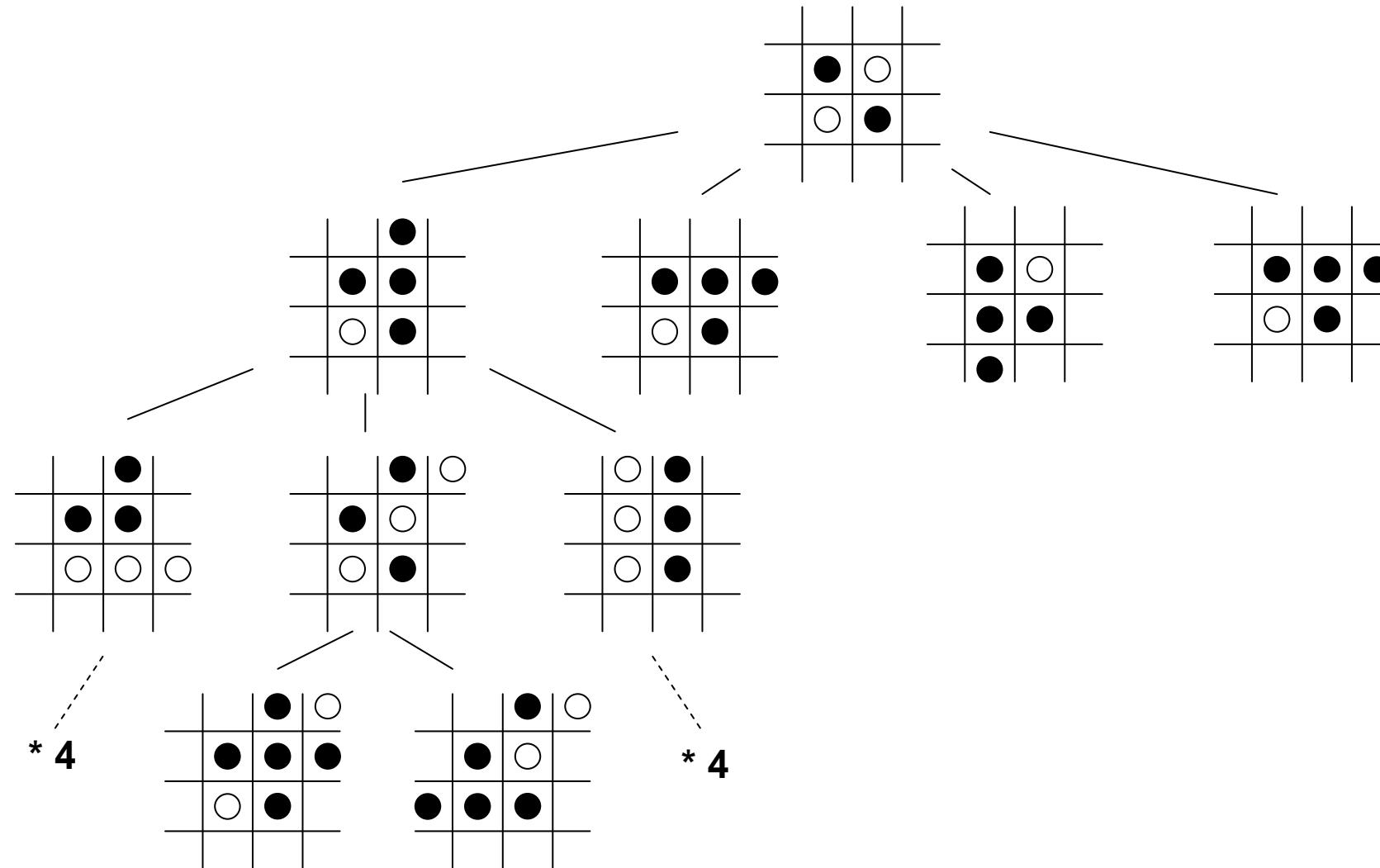


# Критериуми за создавање на дрвото

- Почетна состојба: Скопје
- Крајна цел: Скопје
- Услов: трговскиот патник мора да помине низ сите градови
  
- Одново не е дефинирана цената на транспортот меѓу две места
- Листата градови со кои има врска еден град се подредени по азбучен ред
- Доколку постои цена, овој критериум може да се промени



# Играње игри: реверси / отело





# Критериуми за создавање на дрвото на играта

- Почетна состојба: два пари пулови
- Цели:
  - таблата е пополнета
  - не може да се одигра потег
- Услов: секој играч наизменично става само еден пул на празно место со кое заробува еден до два противнички пула
- Менусостојба: по ставањето нов пул, заробените пулови ја добиваат неговата боја



# Интегрирање

$$\int \frac{dx}{\cos^2 x}$$
$$u = \frac{1}{\cos^2 x} = \sec^2 x$$
$$dv = dx$$
$$x \sec^2 x - \int x \tan(x) dx$$
$$\tan(x)' = \sec^2 x$$
$$\tan(x) + C$$

# Критериуми за интеграцијата

- Почетна состојба:  $\int \frac{dx}{\cos^2 x}$
- Крајна состојба: интегралот е решен, т.е не постои подинтегрална вредност што треба да се замени
- Дозволени оператори: таблични интеграли, тригонометриски функции, парцијален извод

# Расудување

- Знаејќи дека:
  - Сите живи суштства растат.
  - Луѓето се живи суштства.
  
- Да се докаже (провери) точноста на:
  - Сите луѓе растат.

# Критериуми за расудување

- Почетна состојба: база факти
  - Цел: потенцијален факт
  - Оператори: правила за расудување
- 
- Расудувањето ќе биде детално разработено во текот на април 2015

# Синтаксичка обработка

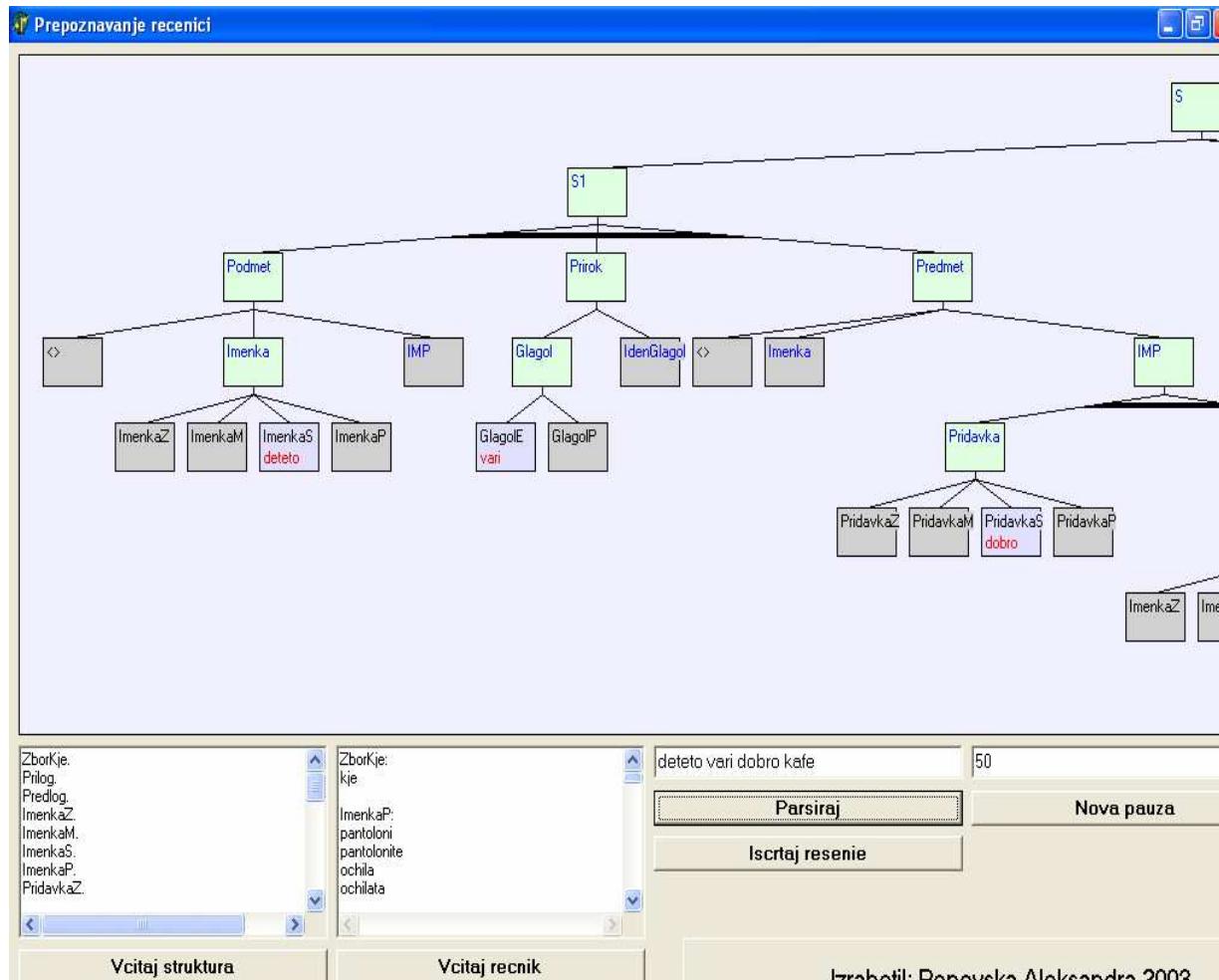
- Да се одреди синтаксичката исправност на речениците:
  - Времето е сончево и пријатно.
  - Времето сончево пријатно е.
  - Сончевото време е пријатно.
  - Сончево и пријатно време.

# Критериуми за парсирањето

- Почетна состојба: S (нетерминален симбол)
- Крајна цел: терминални симболи кои одговараат на зборовите во реченицата.
- Оператори: правила за трансформација на нетерминалите во нетерминални и терминални симболи



# Парсирање во И/ИЛИ дрво



# Типови проблеми

- Детерминистички проблеми или проблеми во кои може да се набљудува
  - агентот точно знае во која состојба може да стигне
- Конформни или проблеми кои не може да се набљудуваат
  - агентот нема идеја каде се наоѓа
- Проблеми на контингенција
  - недетерминистички проблеми со можност за делумно набљудување
- Истражувачки проблеми, т.е проблеми во кои не е познат просторот на преバラување



# Како формално се дефинира проблемот на пребарувањето

Секој детерминистички проблем има:

1. Почетна состојба (на пр. Скопје)
2. Акции или состојби на премин (соседен град до кој има автопат)
3. Крајна цел
  - Експлицитна: Охрид
  - Имплицитна: победа во играта реверси
4. За проблемите сврзани со пат, дефинирана е и цена на патот (цена на картата, должина во километри и сл.)

# Основна поделба на пребарувањето

- Неинформирано пребарување (uninformed search)
  - на слепо (blind search)
  - со груба сила (brute-force search)
  - неинформирано (uninformed search)
  - наивно пребарување (naïve search)
- Информирано пребарување (informed search)
  - локално пребарување (local search)
  - хевристичко (heuristic или metaheuristic)

# Неинформирано пребарување

1. По прва широчина (прво по широчина)
2. По прва длабочина (прво по длабочина)
3. Со изедначена (унифицирана) цена
4. Со ограничена длабочина
5. Со итеративно одење во длабочина
6. Двонасочно

# Информирано пребарување

1. Најдобриот прв (прво најдобриот)
2. A\* пребарување
3. Хевристички
4. Локално пребарување
5. Искачување по рид

# Основна стратегија

- Дефинирај го дрвото на пребарување, развивајќи (разложувајќи) ги постепено неговите јазли
- Кај слепото пребарување, кое е неинформирано, при развивањето не се елиминираат патиштата што делуваат ненадежно, оти сите се еднакво надежни
- Ова предизвикува комбинаторна експлозија, но затоа постои и информирано пребарување



# Дали просторот на преbaraување е голем?



# Критериуми за вреднување на пребарувањето

- Комплетност, т.е. дали секогаш го наоѓа решението доколку тоа постои
- Временска комплексност (сложеност): број на јазлите кои се генерираат при пребарувањето
- Просторна комплексност (сложеност): најголем број јазли во меморијата
- Оптималност: дали секогаш го наоѓа решението со најниска цена

# Мерки на комплексноста

- b: најголемиот фактор на разгранување (branching factor)
- d: длабочина на решението со најниска цена (depth)
- m: максимална длабочина на просторот на пребарување (maximum depth)



# Прашања?