

Основне академске студије
Информациони инжењеринг

Методе и технике науке о подацима

Логичко програмирање у језику Prolog

(материјали за вежбе)

Језик *Prolog*

језик за логичко програмирање

заснива се на примени логике и раду над дефинитним клаузама

декларативно оријентисан

процедурални аспекти битни

динамички типизован

настанак језика

франц. *PROgrammation en LOGique*

Марсеј, Француска

почетак 70-их година XX века

Језик *Prolog* – подаци

терм

- терм као представа података

 - структура података

- врсте термова

 - променљива

 - атомички терм (константа)

 - сложени терм (структура)

Терм – врсте термова

променљива

променљива као представа неодређеног ентитета

назив променљиве

низ знакова (велика и мала слова, цифре, доња црта) који почиње великим словом или доњом цртом

анонимна променљива представљена знаком _

пример

променљива X

променљива NazivDeLa

Терм – врсте термова

атомички терм (константа)

врсте атомичких термова

атом

низ знакова (велика и мала слова, цифре, доња црта) с почетним малим словом

пример

атом `novela`

низ знакова између апострофа

пример

атом `'roman'`

низ посебних знакова

број

цео број

број с покретним зарезом

...

Терм – врсте термова

сложени терм (структура)

изграђен од функтора и низа аргумената

функтор је атом

низ аргумената је између заграда а аргументи су одвојени запетама

аргумент је терм

подржава угњеждавање

аргумент може бити сложени терм

арност (енгл. *arity*)

број аргумената код сложеног терма

пример

сложени терм `obuhvata(proza, roman)`

арности 2

Језик *Prolog* – појмови

клауза (Хорнова клауза)

врсте клауза

чињеница

правило

упит

база знања

колекција чињеница и правила

Клауза

чињеница

основни облик

R.

чињеница важи

Клауза

правило

основни облик

$$Q \text{ :- } P_1, P_2, \dots, P_n.$$
$$n \geq 0$$

правило изграђено од главе и тела

$$\text{Glava} \text{ :- } \text{Telo}.$$

глава сачињена од циља

тело сачињено од конјункције циљева

глава важи ако важи тело

чињеница као посебан случај правила

Клауза

правило

посебне ознаке

- :-
ако (импликација у обрнутом смеру)
- ,
- и (конјункција)
- ;
- или (дисјункција)

Клауза

упит

основни облик

? - $S_1, S_2, \dots, S_n.$

$n \geq 0$

потребно испитати важење

Радно окружење *SWI-Prolog*

бесплатно радно окружење за језик *Prolog*

доступно за више оперативних система
системи фамилија *Windows, Linux, macOS*

интерактивни рад путем конзоле

екстензија *pl* за изворне датотеке

актуелна верзија 9.2.9-1

Интернет сајт

<https://www.swi-prolog.org/>

Рад с језиком *Prolog*

рад путем конзоле

одзивни симбол

? -

листање резултата извршавања упита

помоћу тастера ;

Рад с језиком *Prolog*

опште наредбе

очитавање путање до радног директоријума

```
pwd.
```

очитавање садржаја радног директоријума

```
ls.
```

подешавање путање до радног директоријума

```
cd('putanja').
```

учитавање програма

```
consult('datoteka').
```

```
[datoteka].
```

датотека с програмом очекивано у радном директоријуму

екстензија *pl* може бити изостављена из назива приликом учитавања

учитавање измењене верзије

```
make().
```

завршетак рада

```
halt.
```

...

Рад с језиком *Prolog*

дебаговање

активирање текстуалног режима праћења

`trace.`

главни режим дебаговања у оквиру конзоле

могуће радње приликом појединачних корака у режиму праћења

наставак извршавања до следеће тачке

тастер `c` или тастер `Space` или тастер `Enter`

прекид извршавања

тастер `a`

...

активирање графичког режима праћења

`gtrace.`

завршетак дебаговања

`nodebug.`

заустављање свих режима дебаговања

...

Рад с језиком *Prolog*

коментари

једноредни коментар

`# komentar`

обухвата садржај од знака `#` па закључно с први следећим знаком за нови ред

вишередни коментар

`/* komentar */`

може обухватити више знакова за нови ред

Пример А – база знања

база знања *v15-književnost.pl*

```
pisac(dante).
```

```
pisac(miloš_crnjanski).
```

```
pisac(ivan_aleksandrovič_gončarov).
```

```
pisac(džejms_džojs).
```

```
napisao(dante,božanstvena_komedija).
```

```
napisao(miloš_crnjanski,seobe).
```

```
napisao(miloš_crnjanski,roman_o_londonu).
```

```
napisao(ivan_aleksandrovič_gončarov,oblomov).
```

*осам клауза:
осам чињеница*

Пример А – припрема за рад

покренути окружење *SWI-Prolog*

подесити окружење за употребу кодног формата *UTF-8* (у случају рада на оперативном систему фамилије *Windows*)

употребити предикат `set_prolog_flag/2`

```
?- set_prolog_flag(encoding,utf8).
```

одабрати прикладан фонт (нпр. *Courier New* или *Consolas*)

мени *Settings* – ставка *Font ...* – поље *Font*:

поставити датотеку *v15-književnost.pl* у одабрани директоријум

подесити кроз окружење путању до радног директоријума тако да одговара одабраном директоријуму

употребити предикат `cd/1`

```
?- cd('putanja').
```

Пример А – интерактивни рад 1/4

```
?- ['v15-književnost'].  
true.
```

```
?- pisac(dante).  
true.
```

```
?- pisac(franc_kafka).  
false.
```

```
?- napisao(dante,božanstvena_komedija).  
true.
```

```
?- napisao(dante,oblomov).  
false.
```

```
?-
```

Пример А – интерактивни рад 2/4

```
?- pisac.
```

```
ERROR: Unknown procedure: pisac/0
```

```
ERROR:         However, there are definitions for:
```

```
ERROR:         pisac/1
```

```
false.
```

```
?- pisac(X).
```

```
X = dante ;
```

```
X = miloš_crnjanski ;
```

```
X = ivan_aleksandrovič_gončarov ;
```

```
X = džeјms_džoјs.
```

```
?-
```

Пример А – интерактивни рад 3/4

```
?- napisao(X,oblomov).
```

```
X = ivan_aleksandrovič_gončarov.
```

```
?- napisao(miloš_crnjanski,X).
```

```
X = seobe ;
```

```
X = roman_o_londonu.
```

```
?- napisao(džejms_džojs,X).
```

```
false.
```

```
?-
```

Пример А – интерактивни рад 4/4

```
?- napisao(X,X).  
false.
```

```
?- napisao(X,Y).  
X = dante,  
Y = božanstvena_komedija ;  
X = miloš_crnjanski,  
Y = seobe ;  
X = miloš_crnjanski,  
Y = roman_o_londonu ;  
X = ivan_aleksandrovič_gončarov,  
Y = oblomov.
```

```
?-
```

Пример Б – база знања

база знања *v15-lingvistika.pl*

```
romanski(italijanski).
```

```
romanski(rumunski).
```

```
slovenski(poljski).
```

```
slovenski(srpski).
```

```
indoevropski(X):-romanski(X).
```

```
indoevropski(X):-slovenski(X).
```

*шест̄ клауза:
чет̄ири чињенице и
два њ̄равила*

Пример Б – припрема за рад

покренути окружење *SWI-Prolog*

поставити датотеку *v15-lingvistika.pl* у одабрани директоријум

подесити кроз окружење путању до радног директоријума тако да одговара одабраном директоријуму

употребити предикат `cd/1`

```
?- cd('putanja').
```

Пример Б – интерактивни рад 1/4

```
?- ['v15-lingvistika'].  
true.
```

```
?- romanski(rumunski).  
true.
```

```
?- slovenski(J).  
J = poljski ;  
J = srpski.
```

```
?-
```

Пример Б – интерактивни рад 2/4

```
?- indoevropski(J).
```

```
J = italijanski ;
```

```
J = rumunski ;
```

```
J = poljski ;
```

```
J = srpski.
```

```
?- indoevropski(poljski).
```

```
true.
```

```
?- indoevropski(rumunski).
```

```
true ;
```

```
false.
```

```
?-
```

Пример Б – интерактивни рад 3/4

```
?- trace.  
true.
```

```
[trace] ?- indoevropski(rumunski).  
  Call: (8) indoevropski(rumunski) ? creep  
  Call: (9) romanski(rumunski) ? creep  
  Exit: (9) romanski(rumunski) ? creep  
  Exit: (8) indoevropski(rumunski) ? creep  
true ;  
  Redo: (8) indoevropski(rumunski) ? creep  
  Call: (9) slovenski(rumunski) ? creep  
  Fail: (9) slovenski(rumunski) ? creep  
  Fail: (8) indoevropski(rumunski) ? creep  
false.
```

```
[trace] ?- nodebug.  
true.
```

```
?-
```

Пример Б – интерактивни рад 4/4

```
?- gtrace.
```

```
% The graphical front-end will be used for subsequent tracing  
true.
```

```
[trace] ?- indoevropski(rumunski).
```

```
true ;
```

```
false.
```

```
[trace] ?- nodebug.
```

```
true.
```

```
?-
```

Извршавање

коришћење поступка СЛД резолуције

С – селекција, Л – линеарна, Д – дефинитне клаузе

примена свођења на противречност

креће се од негације тврђења које је предмет интересовања

пример

?- `indoevropski(rumunski)`

што је по значењу као

:- `indoevropski(rumunski)`

односно дато је правило које садржи премису али је закључак празан

примена корака резолуције

за две клаузе формирање њихове резолвенте

потребно показати противречност

потребно извести празну клаузу

Извршавање

спровођење унификације два терма

извођење синтаксне унификације термова током примене поступка резолуције

правила унификације

за два атомичка терма, могуће је спровести њихову унификацију акко су исти
за два терма од којих је један променљива а други је терм произвољне врсте,
могуће је спровести њихову унификацију и терм који је променљива бива
инстанциран на други терм (применљиво и када су оба терма променљиве)

за два сложена терма, могуће је спровести њихову унификацију акко важи све од
следећег

функтори су им исти

арности су им исте

за све одговарајуће аргументе могуће је спровести унификацију
у случају да за два терма није могуће спровести унификацију по неком од
претходних правила, онда за два терма није могуће спровести унификацију

Извршавање

спровођење унификације два терма

замена (супституција)

пресликавање с променљивих на термове

пример

замена M као пресликавање с променљиве X на терм $opis(Y)$

$\{X = opis(Y)\}$

примена замене

примена замене M на терм A

за свако пресликавање $N = n$ из замене M треба да у терму A свака појава променљиве N буде замењена термом n

пример

примена замене $\{X = tekst\}$ на терм $opis(X)$ резултује термом $opis(tekst)$

инстанцирање

терм A представља инстанцу терма B ако постоји замена M таква да се њеном применом на терм B добија терм A

пример

терм $opis(tekst)$ је инстанца терма $opis(X)$

Извршавање

спровођење унификације два терма

провера присуства

провера да ли је променљива која треба бити замењена термом присутна у том терму

ако је присутна не би требало наставити са заменом

ако није присутна требало би наставити са заменом

пример

променљива X и терм `opis(X)`

променљива X присутна у терму `opis(X)`

провера присуства је с теоријског становишта препозната и подржана

провера присуства традиционално се не изводи у језику *Prolog*

погодност су боље перформансе приликом извршавања

мана је што извођење унификације без провере присуства може довести до неисправних закључака

Извршавање

опште напомене

претрага клауза одвија се од горе ка доле

по редоследу задавања клауза

обрада циљева одвија се слева надесно

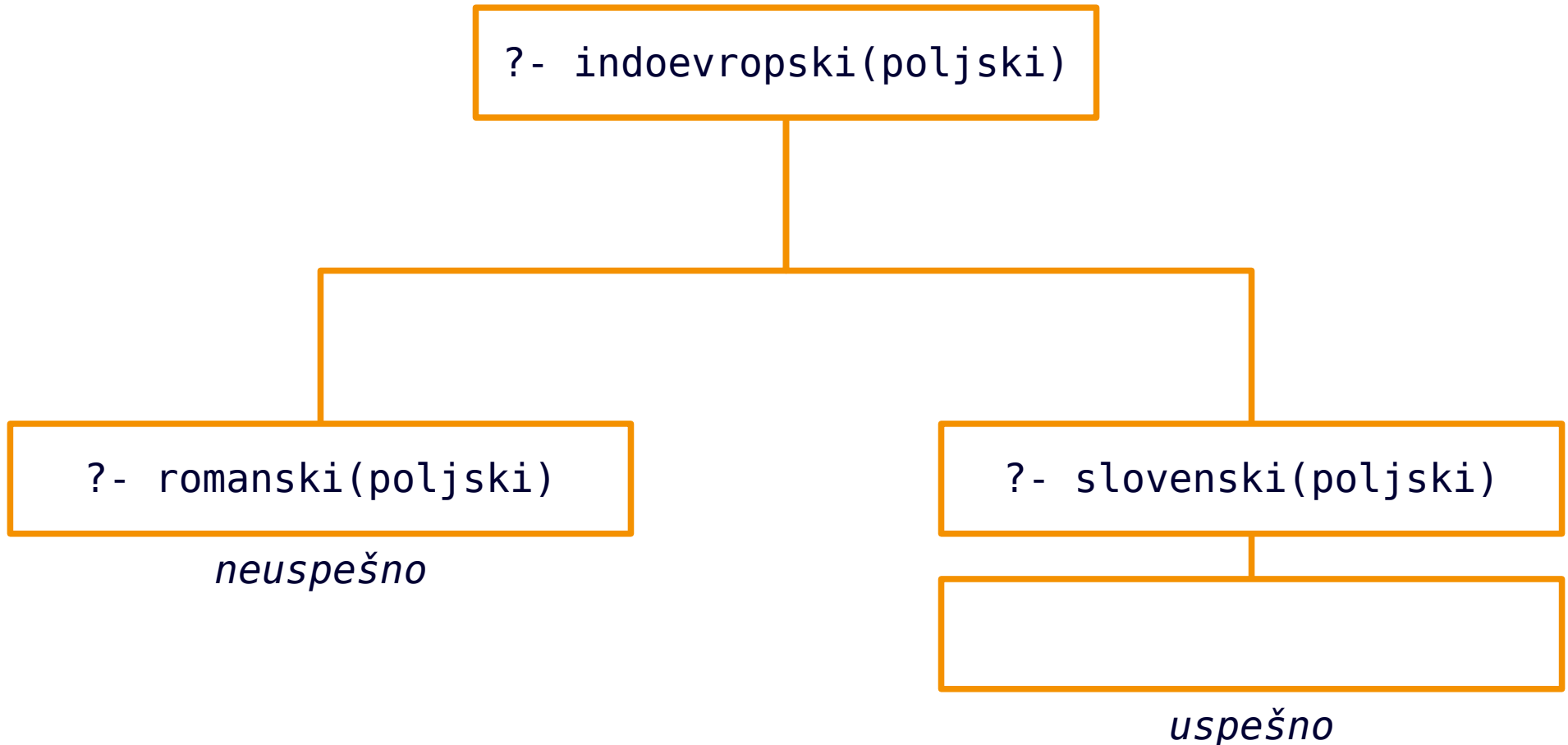
постоји могућност враћања (енгл. *backtracking*)

ако се текући правац потраге за решењем испостави као промашен могуће је прећи на испитивање других праваца потраге за решењем

Пример Б – извршавање – илустрација 1

пример за упит

?- indoevropski(poljski).



Пример Б – извршавање – илустрација 1

пример за упит

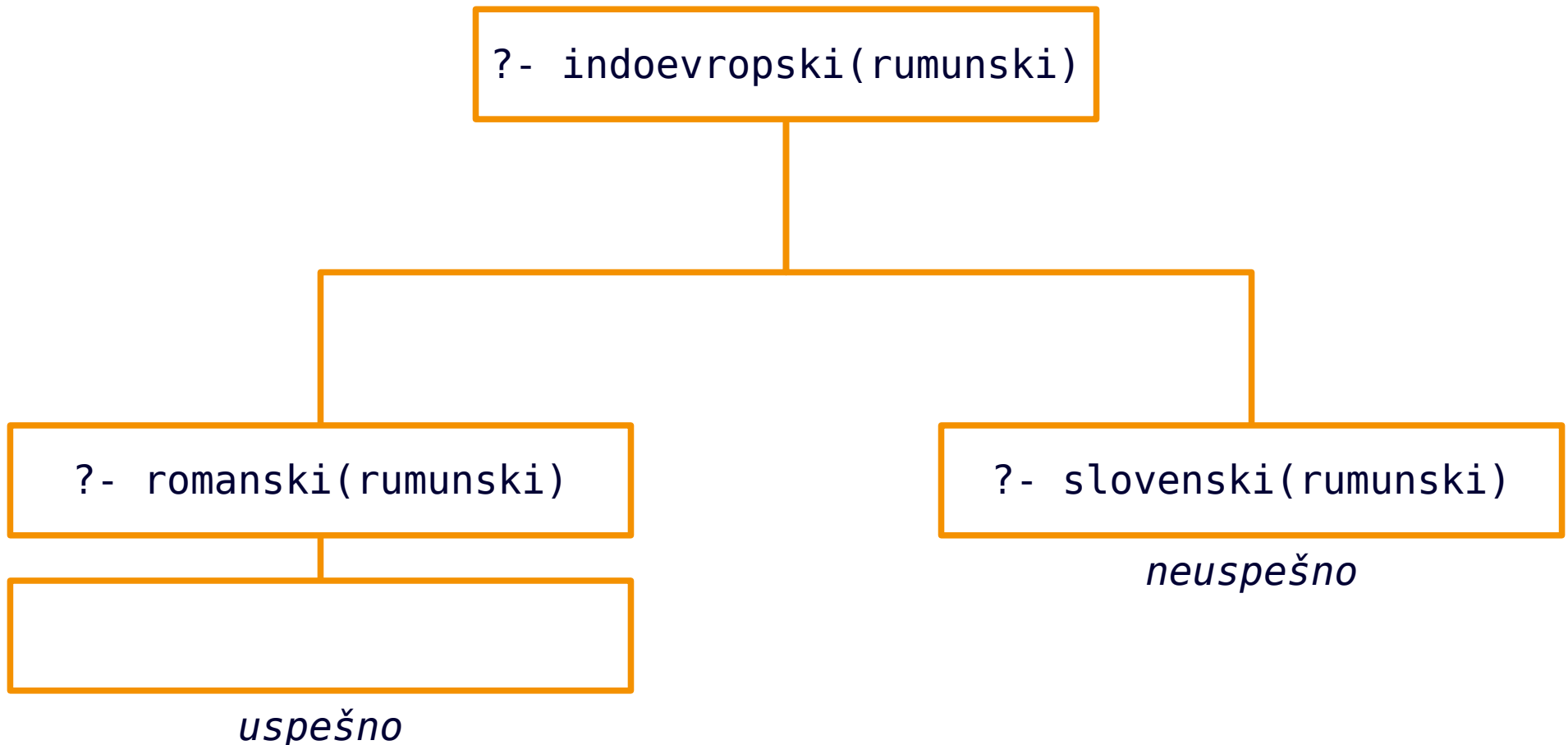
?- indoevropski(poljski).



Пример Б – извршавање – илустрација 2

пример за упит

?- indoevropski(rumunski).

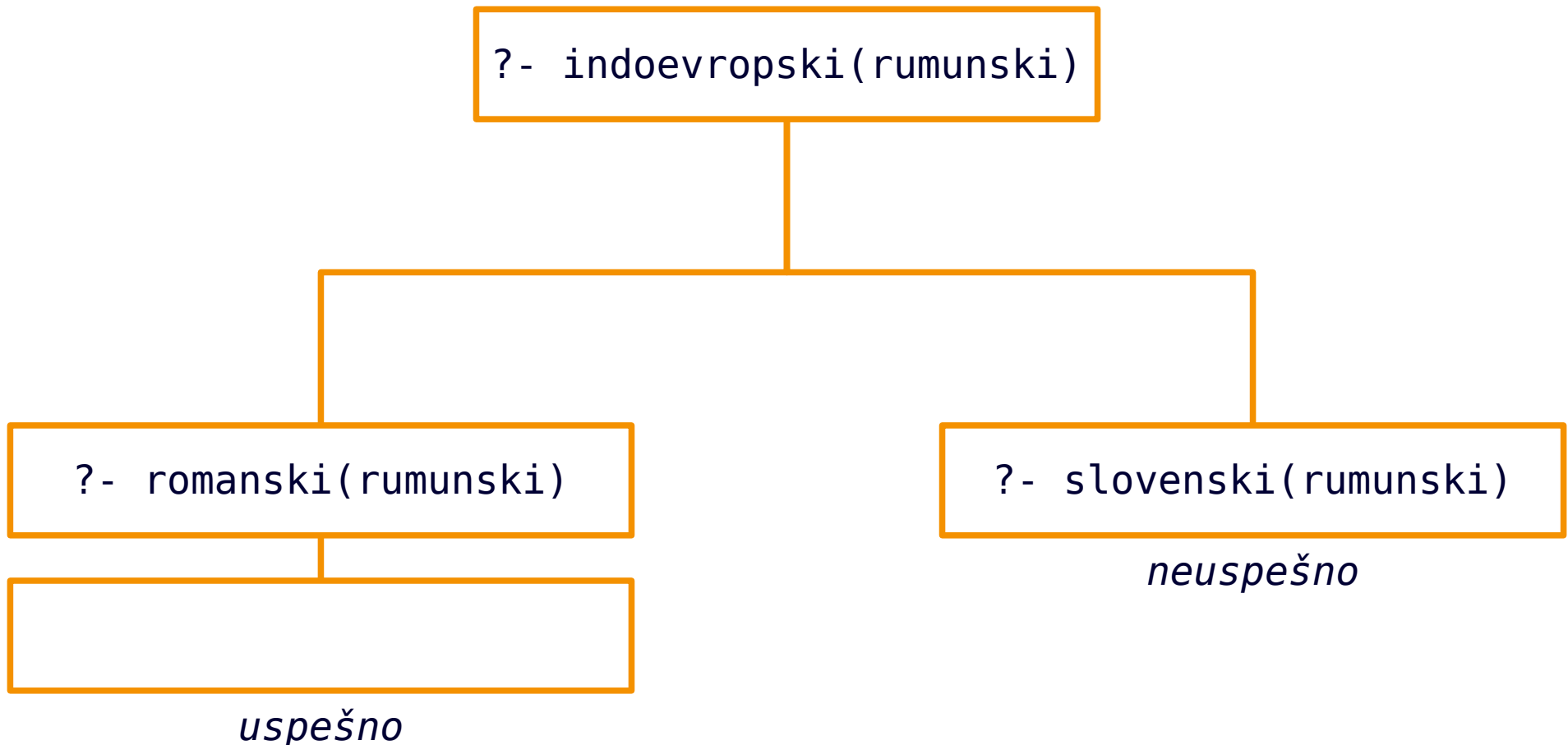


Пример Б – извршавање – илустрација 2

пример за упит

?- indoevropski(rumunski).

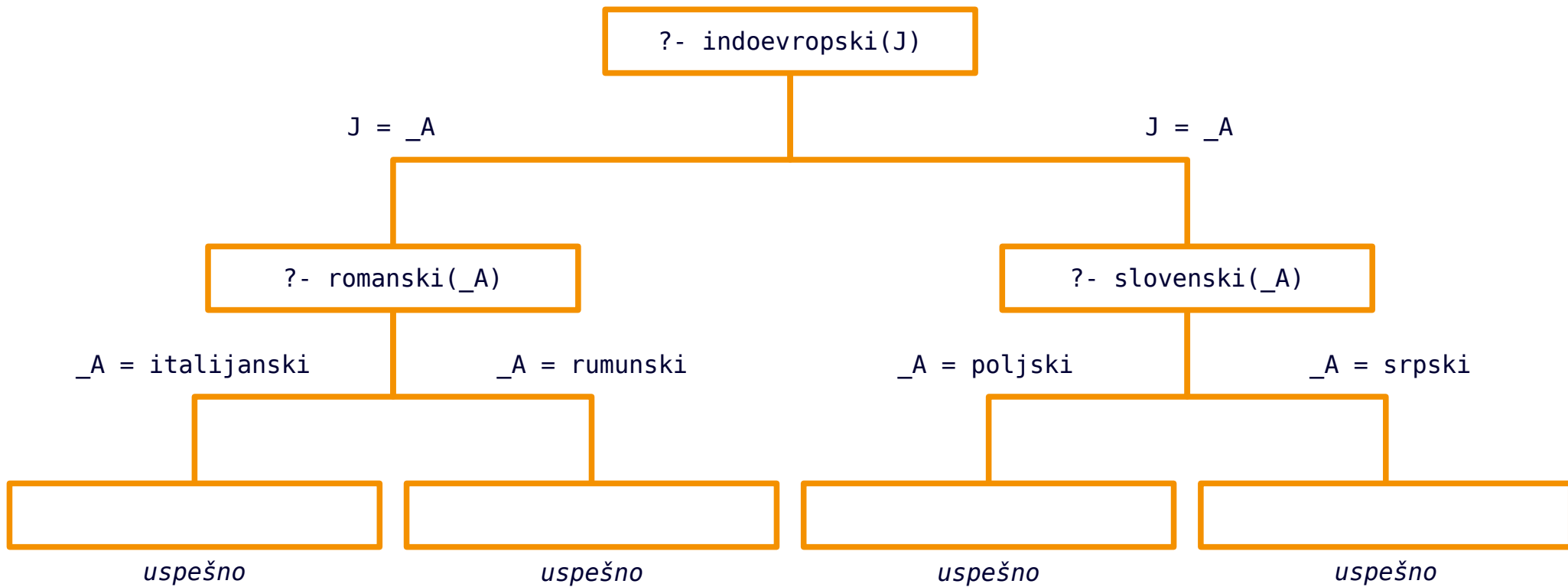
развојем левој подстабла долази се до решења, а наставком, који подразумева враћање и развој десној подстабла, не долази се до додатних решења



Пример Б – извршавање – илустрација 3

пример за упит

`?- indoevropski(J).`



Задатак 1.

Формирати базу знања која ће садржати чињенице о томе где се налази који град. Град је смештен у некој области, а област се налази у некој држави, при чему једна област може бити део неке веће области.

База знања такође треба да садржи правила која на општи начин описују припадност града некој области или држави.

Експериментисати над развијеном базом знања кроз задавање разноврсних упита.

Основна литература

Colmerauer A, Roussel P. The birth of Prolog. History of Programming Languages–II [Bergin TJ, Gibson RG (Eds.)]. ACM Press (New York, NY, USA). 1996; 331–367.

Sterling L, Shapiro E. The Art of Prolog. 2nd edition. MIT Press (Cambridge, MA, USA); 1994.

Flach P. Simply Logical: Intelligent Reasoning by Example. John Wiley & Sons (Chichester, West Sussex, UK); 1994.

Triska M. The Power of Prolog. Internet:
<https://www.metalevel.at/prolog>

Blackburn P, Bos J, Striegnitz K. Learn Prolog Now!. Internet:
<http://lpn.swi-prolog.org/>

SWI-Prolog. SWI-Prolog. Internet: <https://www.swi-prolog.org/>

SWI-Prolog. Manual. Internet:
<https://www.swi-prolog.org/pldoc/refman/>