

Loomuliku keele analüüs ja masintõlkimine

Loogiline programmeerimine ITI0021

Sügis 2016

J.Vain

Ajalooline taust

- Loomuliku keele analüüs oli üks esimesi Prologi rakendusi.
- Keele süntaktiline analüüs (*parsing*) seisneb lause grammatilise struktuuri selgitamises ja selle vastavuse kontrollimises grammatika reeglitele.
- Keele süntees on sõnastiku ja grammatika reeglite abil süntaktiliselt õigete lausete genereerimine.
- Grammatikate formaalse defineerimise süsteemid:
 - Produktsioonid (asendusreeglid)
 - Backus-Nauri kuju (BNF)
 - Definite Clause Grammar (DCG) (vt SWI-Prolog Help)
- jt

Grammatika reeglite esitus

- Produktsioonireegel vasaklineaarses grammatikas:

mitteterminal sümbol → **terminalsümbol**, *mitteterminalsümbol*₁, ... , *mitteterminalsümbol*_n

- DCG reeglid Prologis:

reegli_päis --> reegli_keha

- Näide:
lause --> nimisonafraas, tegusonaafraas.
- Kitsendused reeglite süntaksile Prologis:
 - terminalsümboliks võib olla mistahes Prologi term;
 - mitteterminalsümbol esitatakse Prologis termiga, milleks ei tohi olla list

Näide terminal- ja mitteterminalsümbolid

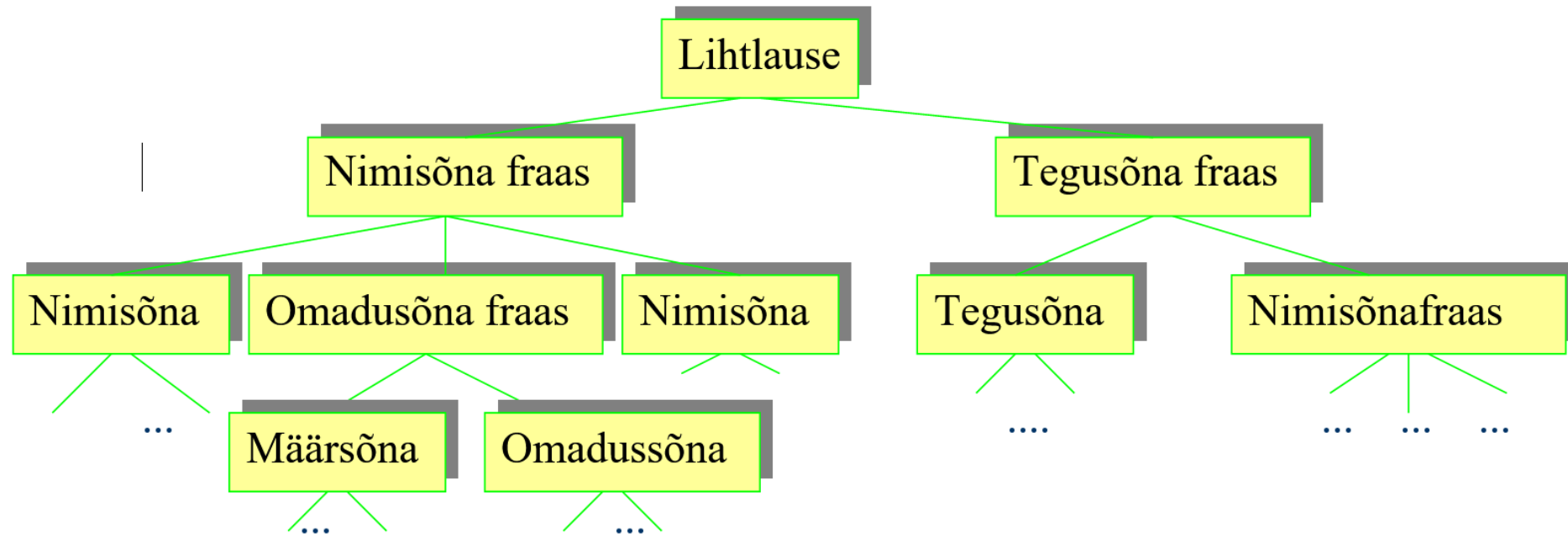
- Reegel Prologis:

```
nimisona --> [poiss];[hobune];[tüdruk]
```

- Reegli tähendus:

Sõnad "poiss", "tüdruk" ja "hobune" on nimisõnad.

Näide (eesti keele lihtlause fraasistruktuuri grammatika)



Näide

- Olgu lause

„Päkapiku liiga napp habe tingib tema sobimatuse jõuluvanaks.“

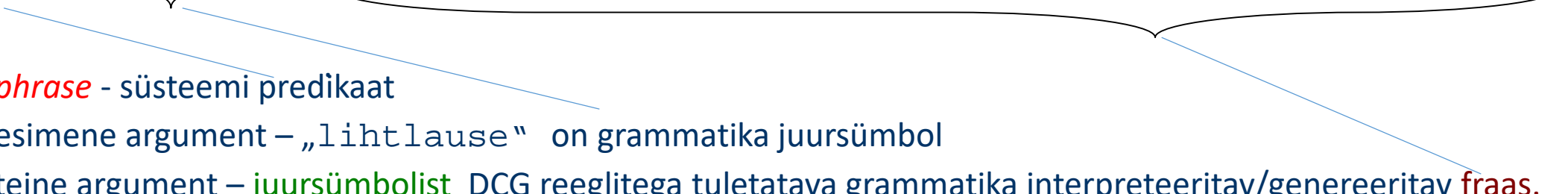
- Defineerime selle lause DCG grammatika.

- lihtlause --> nimisonafraas, tegusonaafraas.
- nimisonafraas --> nimisona, omadussonafraas, nimisona.
- nimisonafraas --> nimisona, nimisonafraas ; [].
- nimisona --> [pakapiku];[habe];[tema];[sobimatuse];[jouluvanaks].
- % terminalsümbolid esinevad reeglis paremal pool ühiklistidena
- omadussonafraas --> maarsona, omadussona.
- maarsona --> [liiga].
- omadussona --> [lyhike].
- tegusonaafraas --> tegusona, nimisonafraas.
- tegusona --> [tingib] ; [pohjustab].

Näide (järg)

- Päring:

?- phrase(lihtlause, [pakapiku, liiga, napp, habe, tingib, tema, sobimatuse, jouluvanaks]).



- *phrase* - süsteemi predikaat
- esimene argument – „lihtlause“ on grammatika juursümbol
- teine argument – juursümbolist DCG reeglitega tuletatava grammatika interpreteeritav/genereeritav fraas.

⌘ vt. swi-prolog samples DCGXPAND


```
lihtlause(A, B) :-  
    nimisonafraas(A, C),  
    tegusonaafraas(C, B).
```

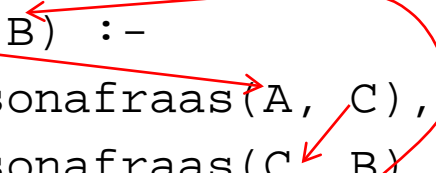
```
nimisonafraas(A, B) :-  
    nimisona(A, C),  
    omadussonafraas(C, D),  
    nimisona(D, B).
```

```
nimisonafraas(A, B) :-  
    (    nimisona(A, C),  
        nimisonafraas(C, B)  
    ;   A=B          ).
```

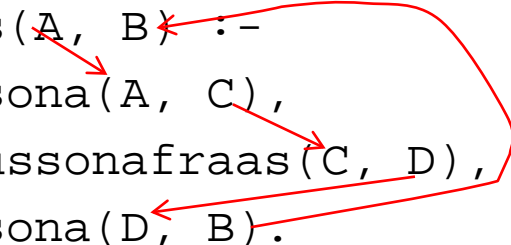
Parsimine: DCG-reeglid prologi programmina

- Parsimiseks genereerib Prolog *DCG* reeglite põhjal programmi

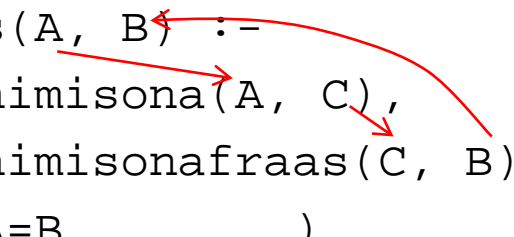
```
lihtlause(A, B) :-  
    nimisonafraas(A, C),  
    tegusonafraas(C, B).
```



```
nimisonafraas(A, B) :-  
    nimisona(A, C),  
    omadussonafraas(C, D),  
    nimisona(D, B).
```



```
nimisonafraas(A, B) :-  
    (    nimisona(A, C),  
        nimisonafraas(C, B)  
    ;   A=B          ).
```



DCG-reeglid prologi programmina

```
tegusonafraas(A, B) :-  
    tegusona(A, C),  
    nimisonafraas(C, B).  
  
nimisona([pakapiku| B], B) .  
nimisona([habe| B], B) .  
nimisona([tema| B], B) .  
nimisona([sobimatuse| B], B) .  
nimisona([jouluvanaks| B], B) .  
  
omadussonafraas(A, B) :-  
    maarsona(A, C),  
    omadussona(C, B).  
  
omadussona([napp|A], A).  
maarsona([liiga|A], A).  
tegusona([tingib|A], A).
```

Märkus: Tõlke muudab mugavaks
Prologi listide pea-saba eraldaja
"|"

sentence →	nounphrase, verbphrase.	
nounphrase →	determiner, nounexpression.	} Alternatiivsed asendused samale mitteterminalile
nounphrase →	nounexpression.	
nounexpression →	noun.	
nounexpression →	adjective, nounexpression.	
verbphrase →	verb, nounphrase.	
determiner →	[the] ; [a].	
noun →	[dog] ; [bone] ; [mouse] ; [cat].	
verb →	[ate] ; [chases].	
adjective →	[big] ; [brown] ; [lazy].	

DCG sisekujus kasutatakse erisuslisti (*difference list*) listide paar X - Y , kus listis Y on listi X sabaosa, mis ei unifitseerunud antud reeglis määratud termiga.

```
?- phrase(sentence, [dog, chases, cat]).

1-1 CALL phrase(sentence, [dog, chases, cat])
  2-1 CALL nounphrase([dog, chases, cat], _0)
    3-1 CALL determiner([dog, chases, cat], _0)
    3-1 FAIL determiner([dog, chases, cat], _0)
  2-1 REDO nounphrase([dog, chases, cat], _0)
    3-1 CALL nounexpression([dog, chases, cat], _0)
      4-1 CALL noun([dog, chases, cat], _0)
      4-1 EXIT noun([dog, chases, cat], [chases, cat])
    3-1 EXIT nounexpression([dog, chases, cat], [chases, cat])
  2-1 EXIT nounphrase([dog, chases, cat], [chases, cat])
  2-2 CALL verbphrase([chases, cat], [])
    3-1 CALL verb([chases, cat], _4)
    3-1 EXIT verb([chases, cat], [cat])
    3-2 CALL nounphrase([cat], [])
      4-1 CALL determiner([cat], [])
      4-1 FAIL determiner([cat], [])
    3-2 REDO nounphrase([cat], [])
      4-1 CALL nounexpression([cat], [])
        5-1 CALL noun([cat], [])
        5-1 EXIT noun([cat], [])
      4-1 EXIT nounexpression([cat], [])
    3-2 EXIT nounphrase([cat], [])
  2-2 EXIT verbphrase([chases, cat], [])
1-1 EXIT sentence([dog, chases, cat])
```

Keele lausete genereerimine

Päringu "phrase/2" ainuke väärtustatud parameeter on grammatika juursümbol

Näide:

Väärtustamata argument

?- phrase(lihtlause, L).

L = [pakapiku, liiga, napp, pakapiku, tingib, pakapiku, liiga, napp, pakapiku] ;

L = [pakapiku, liiga, napp, pakapiku, tingib, pakapiku, liiga, napp, habe] ;

L = [pakapiku, liiga, napp, pakapiku, tingib, pakapiku, liiga, napp, tema] ;

L = [pakapiku, liiga, napp, pakapiku, tingib, pakapiku, liiga, napp, sobimatuse] ;

L = [pakapiku, liiga, napp, habe, tingib, tema, sobimatuse, jouluvanaks].

Masintõlkimine

- Lihtne sarnase grammatikaga, erinevate sõnastikega keelte vahel
- Tõlkeprogrammi üldstruktuur:

```
translate(Lähte_keeles_lause):-
```

```
    find_unknowns(Lähte_keeles_lause).    % kas on tundmatuid sõnu
```

```
translate(Lähte_keeles_lause):-
```

```
    sentence(Lähte_keeles_lause,[], Sihtkeeles_lause,[],),
```

```
    send_print(Sihtkeeles_lause).
```

```
translate(_):-
```

```
    write('Sorry, do\t recognize this type of sentence'),nl.
```

Tõlkeprogrammi üldstruktuur (järg)

```
sentence(Lahtek0, Lahtek, Sihtk0, Sihtk,):-  
    noun_phrase_grp(...),  
    verb_phrase_grp(...).
```

...

Masintõlkimine

- Tõlge erineva grammatikaga (mitte ainult sõnastikega) keelte vahel nõuab lisaks konteksti tundlikku ja semantilist analüüsi.
- Näide: lihtsate sünonüümide tuvastus:
 - `verb(vaata, [look | X], X).`
 - `verb(vaata, [look, around | X], X).`
 - `verb(lõpeta, [end | X], X).`
 - `verb(lõpeta, [quit | X], X).`
 - `verb(lõpeta, [close | X], X).`
- Üldiselt eeldab tõlge konteksti tuvastust ja tõlget fraasi-tasandi reeglitega
- fraasi-tasandi reeglid võivad omada sihtkeele erinevate vastete jaoks mitmeid alternatiive ja grammatilist struktuuri.

Fraasi tuvastuse sidumine reaktsiooniga

Tõlge kontrolleri käskude ja tegevus-stsenaariumite keelde:

- Verbaalsele väljendile võib vastavusse seada reaktsiooni (parametriseeritud programmi), mis genereeritakse ja käivitatakse fraasi interpreteerimise tulemusena.
- Täitva programmi saab genereerida genereeriva grammatikaga, mille konkreetset reeglid väärtustatakse parsimisel kasutatud reeglite abil.
- Näide:

```
? - cmd[ vii, tass, lauale, Y).
```

```
cmd(Command_i, Rest):-
```

```
    phrase(cmd_i_grammar, Command_i, Rest),
```

```
    reaction(action_grammar, Result), !.
```

```
...
```

```
verb --> [vii];[too];... % kui käsus sisaldub „vii“, siis genereeritakse  
                                tegevusgrammatika reegel, mis konkretiseerib “vii”
```

```
verb(A,C):- vii(A,C), assert(action --> vii,...).
```

- Lisalugemist:

- <http://www.learnprolognow.org/slides/official/LPNchapter7.pdf>
- <http://www.learnprolognow.org/slides/official/LPNchapter8.pdf>