

## Loomuliku keele analüüs ja masintõlkimine

- 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:

*mitteterminaal sümbol* → **terminaal sümbol**, *mitteterminaal<sub>1</sub>*, ... , *mitteterminaal<sub>n</sub>*

- DCG reeglid Prologis:

*päis --> keha*

### Näide:

`lause --> nimisonafraas, tegusonaafraas.`

### Kitsendused reeglite süntaksile Prologis:

- terminaal sümboliks võib olla mistahes Prologi term;
- mitteterminaal sümbol esitatakse Prologis termiga, milleks ei tohi olla list.

Näide:

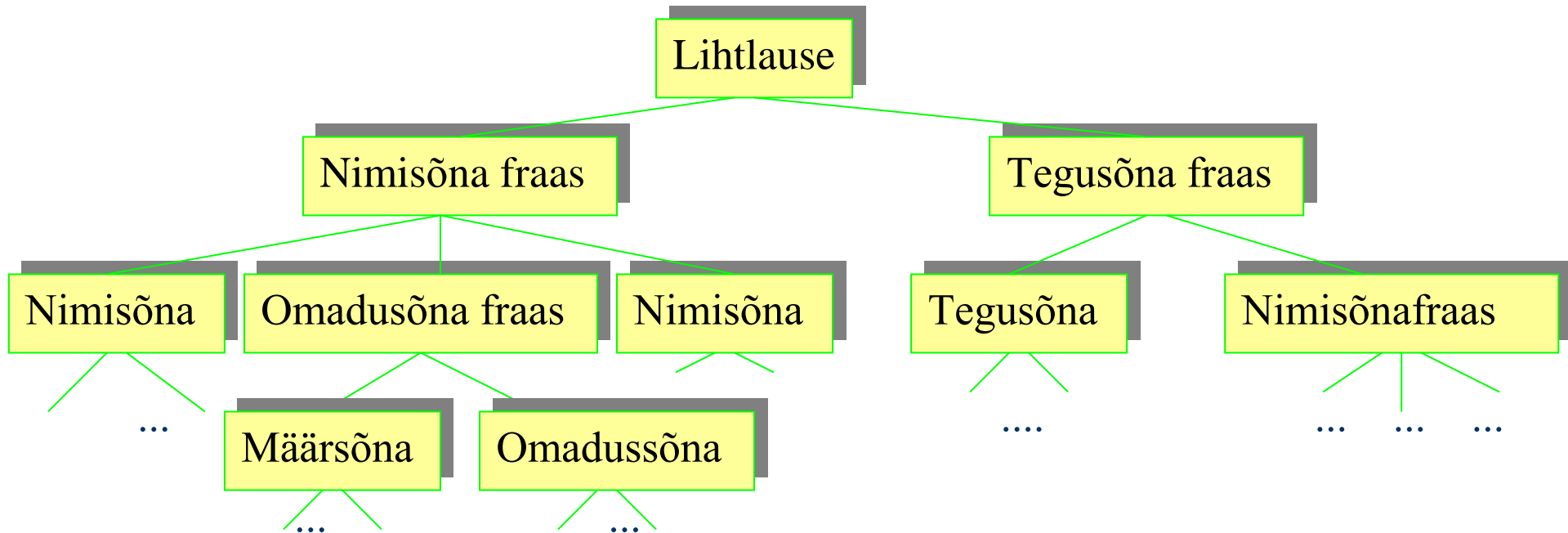
Reegel Prologis:

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

Reegli tähendus:

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

Näide (eesti keele lihtlause fraasistruktuuri grammatika):



Näide:

Päkapiku liiga napp habe tingib tema sobimatuse jõuluvanaks.

Defineerime Prologis grammatika, milles on näitelause süntaks korrektne.

```
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].
```

### Näite selgitus:

Süsteemne predikaat "*phrase*" kontrollib, kas teise argumendi väärtuseks olev **fraas** on esimese parameetri väärtuseks oleva grammatika juursümbolist DCG reeglitega tuletatav keele lause.

?- **phrase**(**lihtlause**,[pakapiku,liiga,napp, habe, tingib, tema, sobimatuse, jouluvanaks]).  
% vt. samples DCGXPAND

Parsimiseks genereerib Prolog *DCG* reeglite põhjal programmi

Näide (järg): Produktsioonireeglite tõlge Prologi programmiks

```
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      ) .
```

```
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 " | "

Veel näide grammatikast ja süntaksianalüüsist

sentence →	nounphrase, verbphrase.	} Alternatiivsed asendused samale mitteterminalile
nounphrase →	determiner, nounexpression.	
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*) s.o. 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:

?- phrase(**lihtlause**, L).

Väärtustamata  
muutuja



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 (sama grammatikaga, erinevate sõnastikega keelte vahel)

### Tõlkeprogrammi üldstruktuur:

```
translate(Lähte_keelee_lause):-  
    find_unknowns(Lähte_keelee_lause). % kas on tundmatuid sõnu  
translate(Lähte_keelee_lause):-  
    sentence(Lähte_keelee_lause,[], Sihtkeelelause,[],),  
    send_print(Sihtkeelelause).  
translate(_):-  
    write('Sorry, do\'t recognize this type of sentence'),nl.
```

### **kus**

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

...

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 tuvastust ja tõlget fraaside tasandi reeglitega

st

fraaside tasandi reeglid võivad omada sihtkeele erinevate vastete jaoks mitmeid alternatiive.

## Fraasi tuvastuse sidumine reaktsiooniga:

- so tõlge kontrolleri käskude keelde
- verbaalsele väljendile võib vastavusse seada parametrizeeritud programmi, mis käivitatakse fraasi äratundmisel

### Näide:

```
?- cmd([alusta, varsti, uut, mõõtmist, ...], Y).
```

```
cmd(Command_i, Rest) :-  
    phrase(cmd_i_grammar, Command_i, Rest),  
    set_bit(0),  
    switch_off(sensor),  
    wait(Delay),  
    switch_on(sensor).
```

### Muster:

```
Reegli päis      cmd(Verbaalne_fraas, Sufiks) :-
```

Reegli keha verbaalsele fraasile vastavat tegevust realiseeriva programmi käsud

Lisalugemist:

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

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