T-(538|725)-MALV, Málvinnsla
Samhengisfrjáls mállýsing og Prolog

Hrafn Loftsson[1] Hannes Högni Vilhjálmsson[1]

[1]Tölvunarfræðideild, Háskólinn í Reykjavík

September 2007

Outline

Outline

Prolog (1970)

Lesning

- http://en.wikipedia.org/wiki/Prolog
- Appendix A í kennslubók.

Hönnunarforsendur

- Að styðja við málvinnslu (e. Natural Language Processing)
- e. Declarative programming
  - (http://en.wikipedia.org/wiki/Declarative_programming

Prolog (1970)

Lesning

- `http://en.wikipedia.org/wiki/Prolog`
- Appendix A í kennslubók.

Hönnunarforsendur

- Að styðja við málvinnslu (e. Natural Language Processing)
- e. Declarative programming
  - (`http://en.wikipedia.org/wiki/Declarative_programming`

Outline

Definite Clause Grammar (DCG)

- Innifalið í Prolog.
- Styður við táknun á samhengisfrjálsri mállýsingu (CFG) í Prolog.
- Prolog breytir CFG reglunum í Prolog reglur.
- Prolog algrímið sér sjálfkrafa um þáttun (e. parsing) án frekari forritunar.
- Helsti kostur við Prolog m.t.t. til CFG er: Fáar línur af kóða!

Definite Clause Grammar (DCG)

Dæmi

```
Reglur                  Orðasafn
-------------------------------------------------------------
s --> np, vp.           det --> [the].     verb --> [brought].
np --> det, noun.       det --> [a].       prep --> [to].
np --> np, pp.          noun --> [waiter]. prep --> [of].
vp --> verb, np.        noun --> [meal].
vp --> verb, np, pp.    noun --> [table].
pp --> prep, np.        noun --> [day].
```

Athugið að vanalega er notaður virkinn :- til að skilja á milli vinstri og hægri hliðar í Prolog. Þegar um DCG er að ræða þá er --> notaður.

Definite Clause Grammar (DCG)

Prolog search

- Almennt gildir að Prolog athugar hvort staðreynd er sönn eða býr til allar lausnir.
- Í tilviki DCG, þá athugar Prolog hvort setningu er hægt að leiða út frá mállýsingunni eða myndar allar setningar sem mállýsingin stendur fyrir.

Prófun

- Starta Prolog: `"c:\Program Files\pl\bin\plwin"`
- Hlaða inn skrá: `?- consult('cfg.pro').`

Definite Clause Grammar (DCG)

Prolog search

- Almennt gildir að Prolog athugar hvort staðreynd er sönn eða býr til allar lausnir.
- Í tilviki DCG, þá athugar Prolog hvort setningu er hægt að leiða út frá mállýsingunni eða myndar allar setningar sem mállýsingin stendur fyrir.

Prófun

- Starta Prolog: `"c:\Program Files\pl\bin\plwin"`
- Hlaða inn skrá: ?- consult('cfg.pro').

Definite Clause Grammar (DCG)

Athuga hvort strengur er í málinu

```
?- s([the, waiter, brought, the, meal, to , the, table], []).
Yes
?- s([the, waiter, brought, the, meal, of , the, day], []).
Yes
```

Mynda allar (málfræðilega réttar) setningar

```
?- s(L, []).
L = [the, waiter, brought, the, waiter] ;
L = [the, waiter, brought, the, meal] ;
L = [the, waiter, brought, the, table] ;
...
```

Definite Clause Grammar (DCG)

Athuga hvort strengur er í málinu

```
?- s([the, waiter, brought, the, meal, to , the, table], []).
Yes
?- s([the, waiter, brought, the, meal, of , the, day], []).
Yes
```

Mynda allar (málfræðilega réttar) setningar

```
?- s(L, []).
L = [the, waiter, brought, the, waiter] ;
L = [the, waiter, brought, the, meal] ;
L = [the, waiter, brought, the, table] ;
...
```

Þýðing á DCG yfir í Prolog reglur (e. clauses)

- DCG reglan s --> np, vp.
- . . . er þýdd yfir í regluna:
- s(L) :- np(L1), vp(L2).
-
- Dæmi: *the waiter brought the meal*
  - L "matches" [the, waiter, brought the meal]
  - L1 "matches" [the, waiter]
  - L2 "matches" [brought, the, meal]

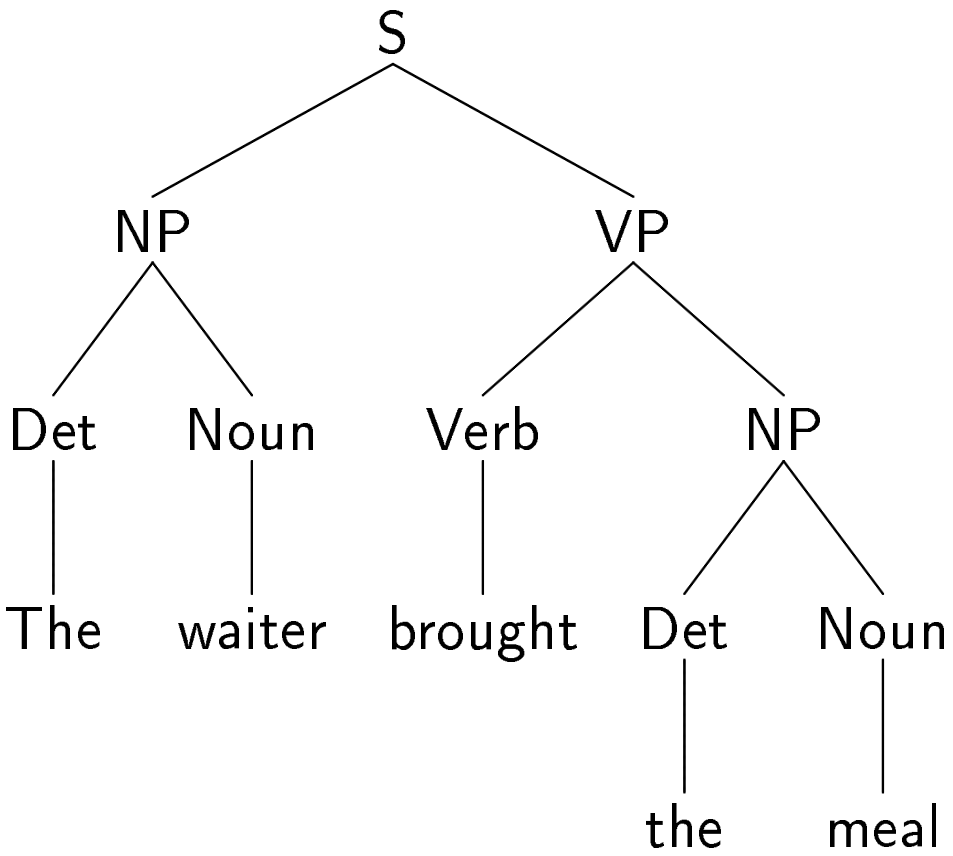
Þýðing á DCG yfir í Prolog reglur (e. clauses)

Til að vera alveg nákvæm þá gerist þetta með svokölluðum *difference lists*

- DCG reglan s --> np, vp.
- . . . er þýdd yfir í regluna:
- s(L1,L) :- np(L1,L2), vp(L2,L).
- DCG reglan: det --> [the].
- . . . er þýdd yfir í staðreyndina:
- det([the | L], L).

Prolog framkvæmir *depth-first search*

- ?- trace.
- ?- s([the, waiter, brought, the, meal], []).



Prolog framkvæmir *depth-first search*

- ?- s([the, waiter, brought, the, meal], []).
- s(L1, L) :- np(L1, L2), vp(L2, L).
  - L1 = [the, waiter, brought, the, meal], L=[].
- np(L1, L) :- det(L1, L2), noun(L2, L).
- det(the | L), L.
  - L = [waiter, brought, the, meal].
  - $\Rightarrow$ L1 = [the].
  - $\Rightarrow$ L2 = [waiter, brought, the, meal].
- noun([waiter, brought, the, meal],L).
- ...

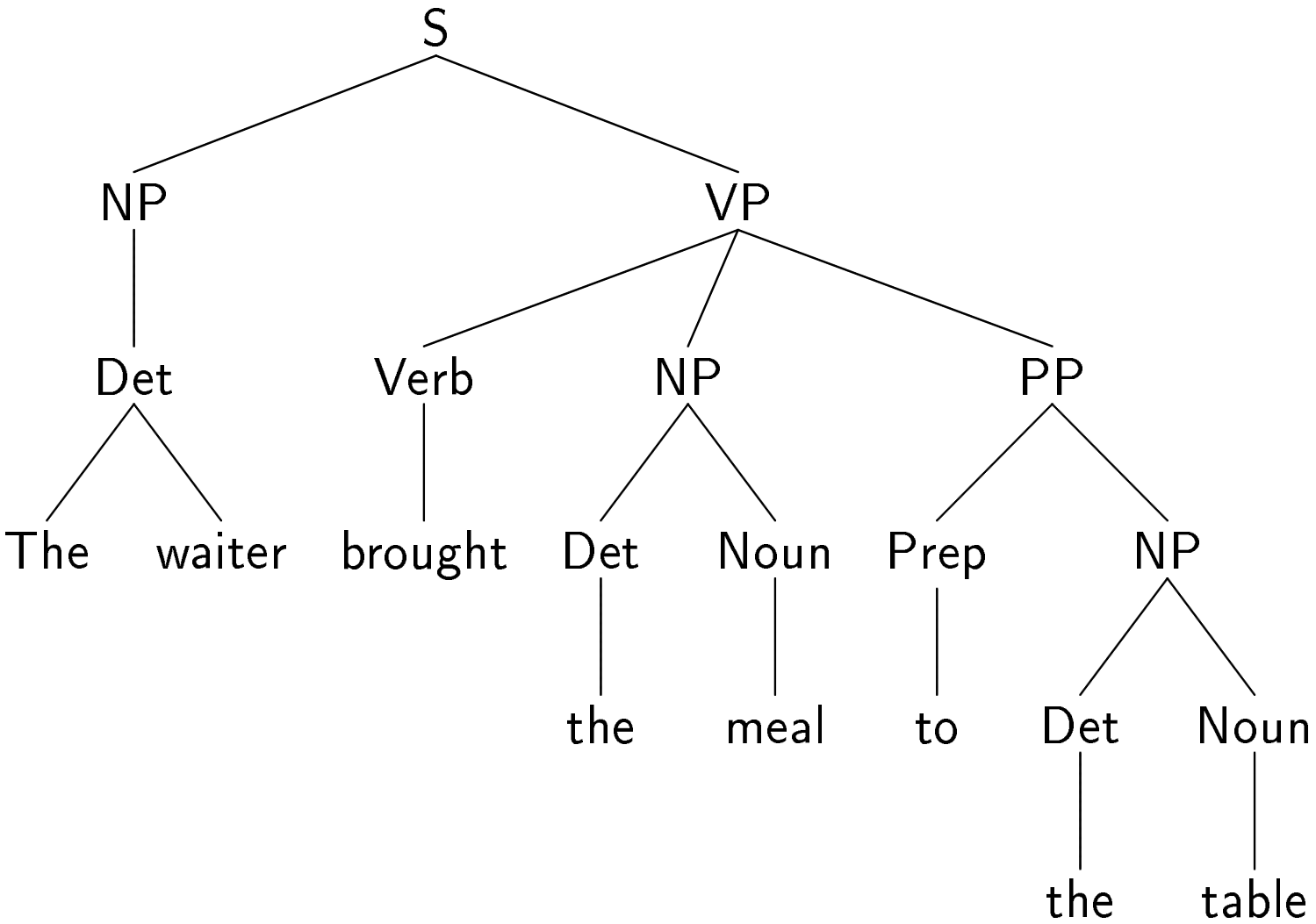
Vinstri endurkvæmni (e. left recursion)

- ?- s([the, brought, the, meal], []).
  ERROR: Out of local stack
- Vegna vinstri endurkvæmni í reglunni np --> np, pp.
- Hægt að eyða vinstri endurkvæmni og það er reyndar nauðsynlegt í Prolog (sjá kafla 8.3.3 í bók).

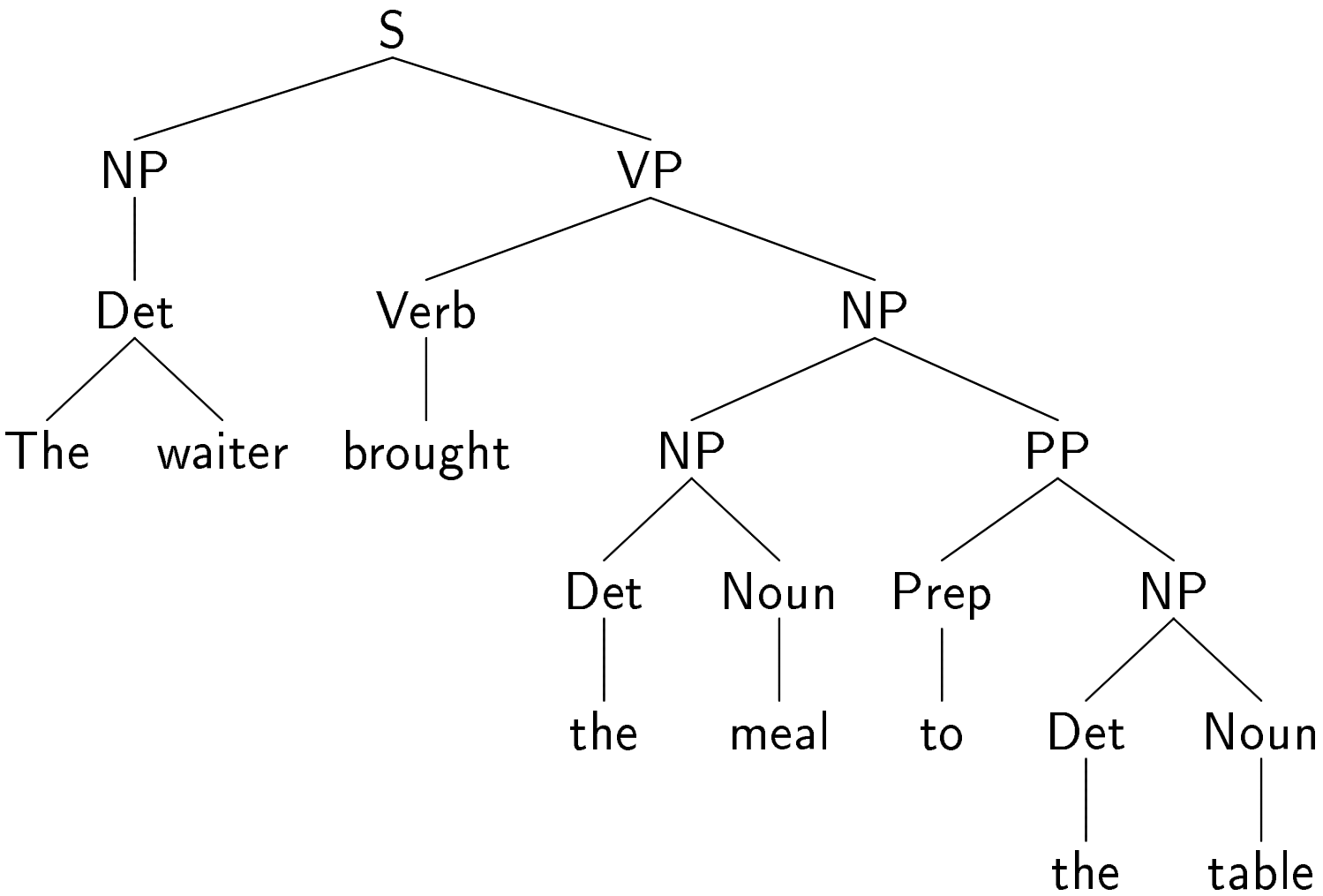
Margræðni í þáttun (e. parsing ambiguity)

- Reglurnar ...
- vp --> verb, np og np --> np, pp
- vp --> verb, np, pp
- ...leiða af sér margræðni, því fleiri en eitt þáttunartré er til fyrir sama strenginn.
- T.d. fyrir setninguna: *The waiter brought the meal to the table*
- Tré 1a og 2a á næstu glærum eru rétt.

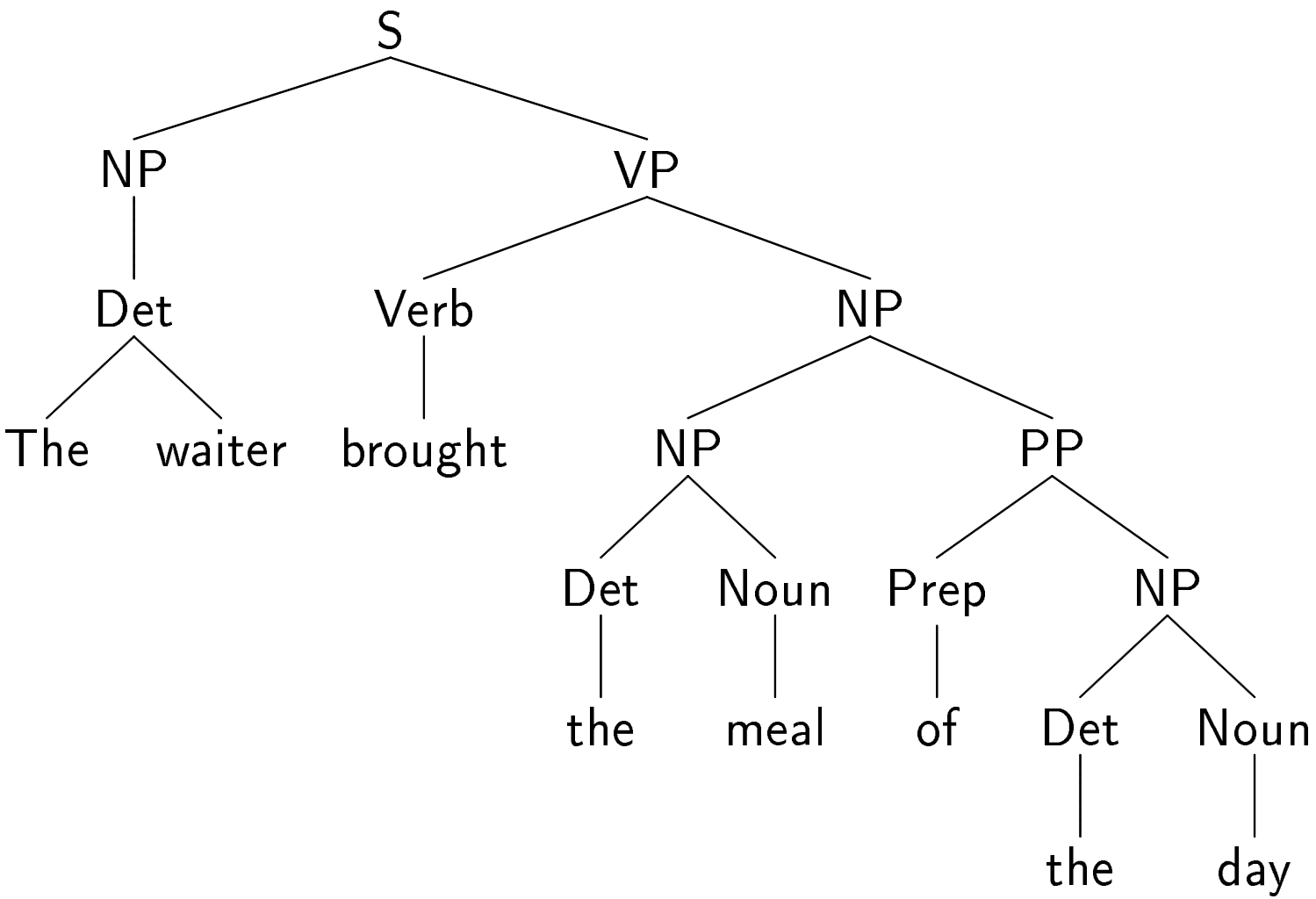
Margræðni í þáttun – Tré 1a



Margræðni í þáttun – Tré 1b



Margræðni í þáttun – Tré 2a



Margræðni í þáttun – Tré 2b