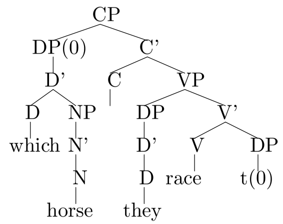
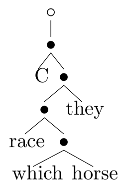
Case marking and concord

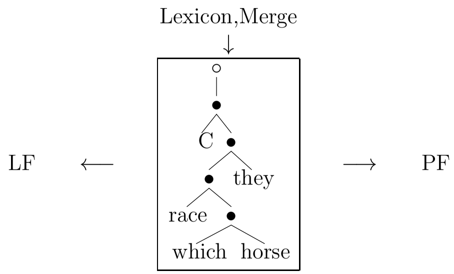
- Background assumptions
  - ⋆ Consensus about language structure
  - ⋆ Performance: some simple models
- Beyond the convergence: Syn-PF interface
  - ⋆ phrasal stress, domain strengthening
  - ⇒ case and concord at PF
  - ⋆ Chinese numbers at PF
  - ⇒ cf 'timing' accts of case and concord

Consensus: Points of convergence

(0) **merge and move**





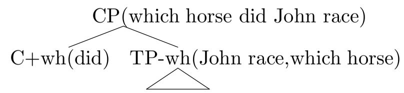


- pronounced, interpreted order ≠ order of leaves derivation tree

. . . other grammatical traditions similar

(1) **generalized CFGs: MCFGs**

[which horse] [did [John race _]]



$$\mathrm{CP}(zyx) \rightarrow \mathrm{C{+}wh}(y)\ \mathrm{TP\text{-}wh}(x,z)$$

(cf. Pollard'84, Seki&al'91, Michaelis'98)

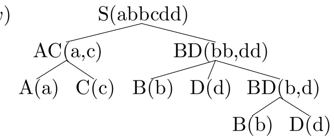
Example 1

S($xzyw$): -AC(x,y), BD(z,w)

AC(xz,yw): -A(x), C(y), AC(z,w)
AC(x,y): -A(x), C(y)

BD(xz,yw): -B(x), D(z), BD(y,w)
BD(x,y): -B(x), D(y)

A(a)
B(b)
C(c)
D(d)



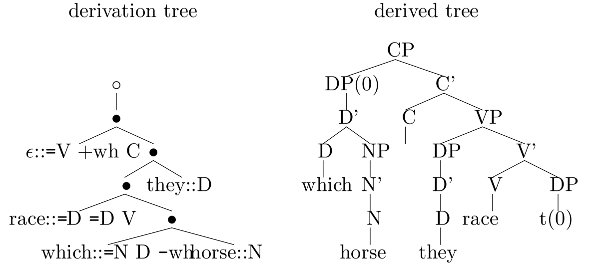
- pronounced, interpreted order ≠ order of derivation tree leaves

(2) **Comparing MGs and generalized CFGs, more carefully**

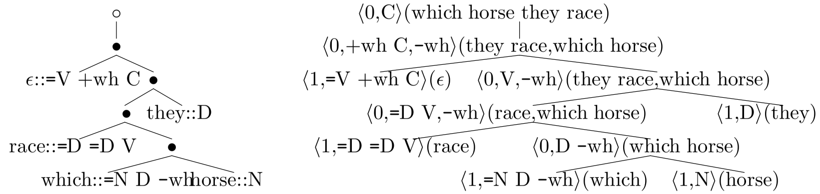
- **Lexical items** associate ph with syn properties:

they::D
race::=D =D V
which::=N D −wh
horse::N
ϵ::=V +wh C

• **Structure** built by merge: external • and internal ∘



• We define a mapping MGs→MCFGs, such that we always have this map between derivations:



• This translation always works – every MG strongly equiv to MCFG

(3) **Thm. 'External convergence'** (Vijay-Shanker,Weir&Joshi'87...)

$$\text{FSG}\subset\text{CFG}\subset\boxed{\boxed{\boxed{\text{TAG=CCG}}\subset\underline{\text{MG=MCFG}}\text{=MCTAG}}\subset\text{PMCFG}}\subset\text{CS}$$

(4) **(MCS)** HLs are 'weakly and strongly' MCS (Joshi'85)

Let's call the largest box in Thm 1, PMCFG, 'approx MCS'

(5) **Thm. 'Internal convergence'** (Michaelis'01,'02;...)

$$\boxed{\text{MG=MGH=MT=DMG=CMG=PMG=RMG=RMCMG=SMMG}}$$

(6) **Consequences**

- ∘ Why are these convergences happening? (Mönnich'10,&al)
- • Performance models (for every grammar) (Stabler'11,&al)
- ⇒ Challenging strict MCS: Case marking

An incremental performance model

(7) TD beam parsing algorithm:

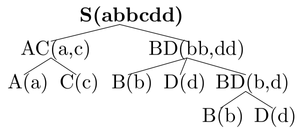
Initialize: Predict root category, insert into beam
While most probable analysis has outstanding predictions:
  If leftmost prediction is terminal:
    check against input; insert result into beam
  Else:
    expand in all possible ways; insert results into beam

(Roark&Johnson'99, Roark'04,...⇒**MCS:** de la Clergerie'02, Mainguy'10, Stabler'11)

TD is a very old idea (cf Frazier'78); now for all MGs, incl remnant movement

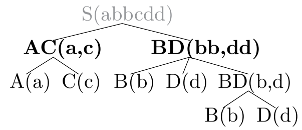
Example 1 continued

Init:



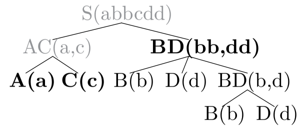
input	memory
abbcdd	**S**

i.



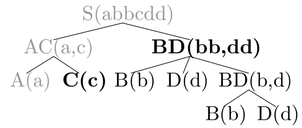
input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$

ii.



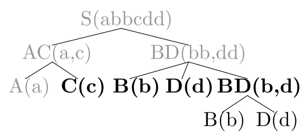
input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$
abbcdd	$\mathbf{A}_{0}\mathbf{BD}_{1,3}\mathbf{C}_{2}$

iii.



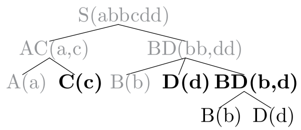
input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$
abbcdd	$\mathbf{A}_{0}\mathbf{BD}_{1,3}\mathbf{C}_{2}$
bbcdd	$\mathbf{BD}_{1,3}\mathbf{C}_{2}$

iv.



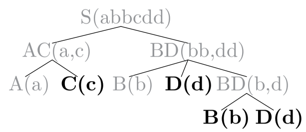
input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$
abbcdd	$\mathbf{A}_0\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{B}_{10}\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$

v.



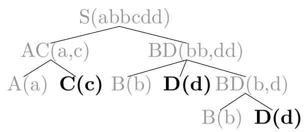
input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$
abbcdd	$\mathbf{A}_0\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{B}_{10}\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$

vi.



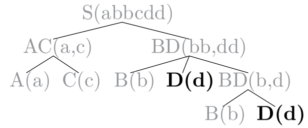
input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$
abbcdd	$\mathbf{A}_0\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{B}_{10}\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{B}_{11}\mathbf{C}_2\mathbf{D}_{30}\mathbf{D}_{31}$

vii.



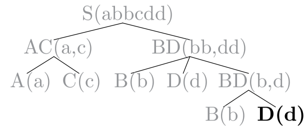
input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$
abbcdd	$\mathbf{A}_0\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{B}_{10}\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{B}_{11}\mathbf{C}_2\mathbf{D}_{30}\mathbf{D}_{31}$
cdd	$\mathbf{C}_2\mathbf{D}_{30}\mathbf{D}_{31}$

viii.



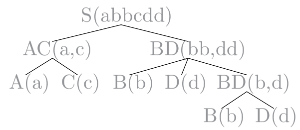
input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$
abbcdd	$\mathbf{A}_0\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{B}_{10}\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{B}_{11}\mathbf{C}_2\mathbf{D}_{30}\mathbf{D}_{31}$
cdd	$\mathbf{C}_2\mathbf{D}_{30}\mathbf{D}_{31}$
dd	$\mathbf{D}_{30}\mathbf{D}_{31}$

ix.



input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$
abbcdd	$\mathbf{A}_0\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{B}_{10}\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{B}_{11}\mathbf{C}_2\mathbf{D}_{30}\mathbf{D}_{31}$
cdd	$\mathbf{C}_2\mathbf{D}_{30}\mathbf{D}_{31}$
dd	$\mathbf{D}_{30}\mathbf{D}_{31}$
d	$\mathbf{D}_{31}$

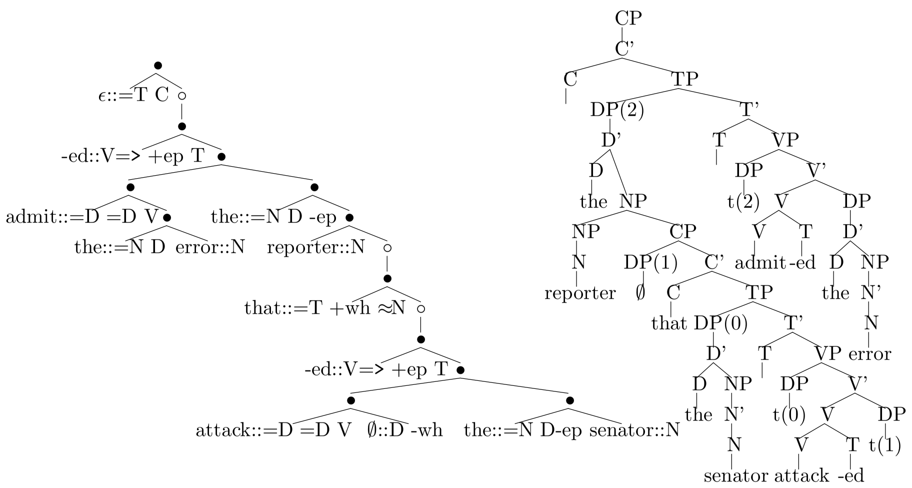
x.



input	memory
abbcdd	**S**
abbcdd	$\mathbf{AC}_{0,2}\mathbf{BD}_{1,3}$
abbcdd	$\mathbf{A}_0\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{BD}_{1,3}\mathbf{C}_2$
bbcdd	$\mathbf{B}_{10}\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{BD}_{11,31}\mathbf{C}_2\mathbf{D}_{30}$
bcdd	$\mathbf{B}_{11}\mathbf{C}_2\mathbf{D}_{30}\mathbf{D}_{31}$
cdd	$\mathbf{C}_2\mathbf{D}_{30}\mathbf{D}_{31}$
dd	$\mathbf{D}_{30}\mathbf{D}_{31}$
d	$\mathbf{D}_{31}$
ϵ	ϵ

- At every point, 1 connected partial tree attached to all predictions (incremental).
- At no point is there any tree structure in $\boxed{\text{parser}}$ memory,
  because every operation is 'local' in a relevant sense (Stabler'10 'finite partition property')
- 2 parts to understand exactly how this model parses your Chomskian syntax:
  - how your MG variant lexicalizes MCFGs (Stabler&Keenan'03; Guillaumin'04)
  - how probabilistic TD applies to these (Stabler'11; Mainguy'10)

Example 2

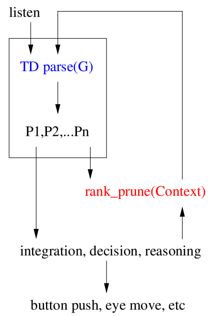


28 node derivation tree and corresponding X-bar tree for King&Just'91 example

1	init.	(trttsa-a-te,	⟨0,C⟩(0,1,2))
2	1.	(trttsa-a-te,	⟨1,=TC⟩(1),⟨0,T⟩(20,21,22))
1	2.	(trttsa-a-te,	⟨0,T⟩(0,1,2))
1	3.	(trttsa-a-te,	⟨0,+epT,-ep⟩(01,1,2,00))
2	4.	(trttsa-a-te,	⟨0,V,-ep⟩(20,21,23,00),⟨1,V=>+epT⟩(22))
3	5.	(trttsa-a-te,	⟨0,D-ep⟩(000,001,002),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
4	6.	(trttsa-a-te,	⟨1,=ND-ep⟩(001),⟨0,N⟩(0020,0021,0022),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
3	7.	(rttsa-a-te,	⟨0,N⟩(0020,0021,0022),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
4	8.	(rttsa-a-te,	⟨1,N⟩(0021),⟨0,≈N⟩(00220,00221,00222),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
3	9.	(ttsa-a-te,	⟨0,≈N⟩(00220,00221,00222),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
3	10.	(ttsa-a-te,	⟨0,+wh≈N,-wh⟩(002201,00221,00222,002200),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
4	11.	(ttsa-a-te,	⟨0,T,-wh⟩(002220,002221,002222,002200),⟨1,=T+wh≈N⟩(00221),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
4	12.	(ttsa-a-te,	⟨0,+epT,-ep,-wh⟩(0022201,002221,002222,0022200,002200),⟨1,=T+wh≈N⟩(00221),⟨0,=DV⟩(20,21,23), ⟨1,V=>+epT⟩(22))
5	13.	(ttsa-a-te,	⟨0,V,-ep,-wh⟩(0022220,0022221,0022223,0022200,002200),⟨1,=T+wh≈N⟩(00221),⟨1,V=>+epT⟩(0022222), ⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
6	14.	(ttsa-a-te,	⟨0,=DV,-wh⟩(0022220,0022221,0022223,002200),⟨1,=T+wh≈N⟩(00221), ⟨0,D-ep⟩(00222000,00222001,00222002),⟨1,V=>+epT⟩(0022222),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
7	15.	(ttsa-a-te,	⟨1,D-wh⟩(002200),⟨1,=T+wh≈N⟩(00221),⟨0,D-ep⟩(00222000,00222001,00222002),⟨1,=D=DV⟩(0022221), ⟨1,V=>+epT⟩(0022222),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
6	16.	(ttsa-a-te,	⟨1,=T+wh≈N⟩(00221),⟨0,D-ep⟩(00222000,00222001,00222002),⟨1,=D=DV⟩(0022221), ⟨1,V=>+epT⟩(0022222),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
5	17.	(tsa-a-te,	⟨0,D-ep⟩(00222000,00222001,00222002),⟨1,=D=DV⟩(0022221),⟨1,V=>+epT⟩(0022222), ⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
6	18.	(tsa-a-te,	⟨1,=ND-ep⟩(00222001),⟨1,N⟩(00222002),⟨1,=D=DV⟩(0022221),⟨1,V=>+epT⟩(0022222),⟨0,=DV⟩(20,21,23), ⟨1,V=>+epT⟩(22))
5	19.	(sa-a-te,	⟨1,N⟩(00222002),⟨1,=D=DV⟩(0022221),⟨1,V=>+epT⟩(0022222),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
4	20.	(a-a-te,	⟨1,=D=DV⟩(0022221),⟨1,V=>+epT⟩(0022222),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
3	21.	(-a-te,	⟨1,V=>+epT⟩(0022222),⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
2	22.	(a-te,	⟨0,=DV⟩(20,21,23),⟨1,V=>+epT⟩(22))
3	23.	(a-te,	⟨1,=D=DV⟩(1),⟨1,V=>+epT⟩(2),⟨0,D⟩(30,31,32))
2	24.	(-te,	⟨1,V=>+epT⟩(2),⟨0,D⟩(30,31,32))
1	25.	(te,	⟨0,D⟩(30,31,32))
2	26.	(te,	⟨1,=ND⟩(1),⟨1,N⟩(2))
1	27.	(e,	⟨1,N⟩(2))
0	28.	(ϵ,	ϵ)

28 step TD recognition of King&Just'91 example

(8) The structure of the model:



P not dependent just on rule, but on tree

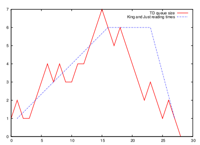
"... what recommends this approach most [is] the ease with which one can estimate string probabilities in a single pass from left to right across the string ... with [standard smoothing] the beam can be greatly narrowed without losing much accuracy, and maintaining complete coverage..." (Roark'04)

(9) We could add: fast, simple, incremental, transparent, possible start for psych model

Assessment of these new simpler models just beginnning...

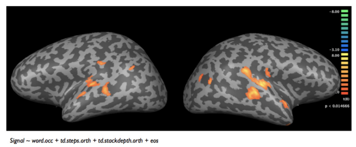
King&Just'91: the reporter that the senator attack -ed _ admit -ed the error
the reporter that _ attack -ed the senator admit -ed the error
(see Example 2 parse, previous page)

object relative



self-paced reading times

story listening



fMRI (vanWagenen, Brennan & Stabler'11)

(10) **Can we expand G in (8) to include all of grammar?**

?Yes Morph: MGs closed wrt FS conditions on trees (Graf'11, Kobele'11)
Phon: MGs closed wrt FS conditions on strings even for OT (Frank&Satta'98, Jäger'02, Heinz'10,...)

?MCS: MCS G defines linguistic structure, weakly and strongly, from perceptual/articulatory structure to syntax to conceptual/intentional structure

No: (i) some G at PF interface; (ii) approx MCS

At the interface: Ph

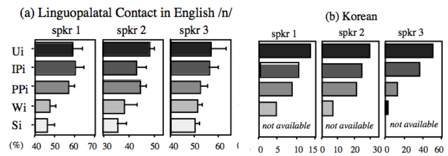
(11) **Boundary effects**

- Whenever John walks # the dog is chasing him
- Whenever John walks the dog # the kids are chasing him
  Whenever[John[walks[the dog]]] [[the kids][are[chasing[him]]]]

(12)

- 89 + 89 + 89 + 89 = a lot
- (89 + 89) × (89 + 89) = a lot
- ...

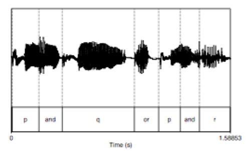
Fougeron&Keating'97 Keating&al'04: /t/ VOT

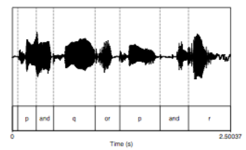


the more brackets, the stronger the segmental effects

'the acoustic properties VOT, total voiceless interval, %voicing during closure, nasal energy minimum, and to a lesser extent stop burst energy and voicing into closure, were found to vary with prosodic position ... They could thus potentially provide cues to listeners about prosodic structure.'

$(p \wedge q) \vee (p \wedge r)$ $((p \wedge q) \vee p) \wedge r$





$p| \wedge q|| \vee p| \wedge r$ $p| \wedge q|| \vee p||| \wedge r$

(Ladd'88;Féry&Truckenbrodt'04;Wagner'05,'07) 'boundaries are recursively scaled in strength'

... (arguably) unbounded # of distinctions, not local

Keating&al comment on the great speaker variability. Although we only see clear initial strengthening and timing effects distinguishing up to 4 levels or so, the effects are not qualitatively different for each level but just strengthening, and so it seems natural to regard this as a recursive process that is applied only to certain depths.

But if # of brackets matters, it is striking that no brackets at all are needed in the TD beam model. If we wanted to mark them in the category system up to a fixed finite bound, we could do that, but it would explode the categories and miss the generalization.

Another thing we could do, instead of marking infinitely many distinctions in the vocabulary, is to put the (relevant) brackets into the spellout, with the idea that $]^n$ is a signal to produce stronger phonetic effects (of whatever kind) than $]^{n-1}$. This would still be MCS, leaving the counting of the brackets to unspecified articulatory and interpretive devices.

It seems much more natural to account for their influence with a module that can see the whole tree – we will offer a proposal of this kind below, and in appendix.

At the interface: Case marking and concord

(13) local and 1-1?
T[~~uCase:N~~][$_{vP}$he[Case:N] v[~~uCase:A~~] sees her[Case:A]]

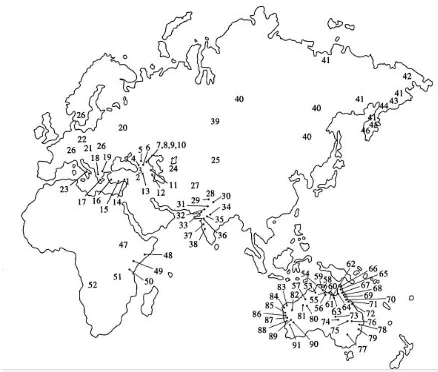
(14) **one-to-many** ('concord')

- der mutmaßliche Täter
  the.N presumed.N perpetrator.N
- des mutmaßlichen Täters
  the.G presumed.G perpetrator.G

(15) **many-to-one**

- thabuju-karra-nguni mijil-nguni (Kayardild: Evans'95,Round'10)
  brother-G-I net-I
  'with brother's net'
  (both studies report Ss won't stack more than 4)
- govel-i igi sisxl-i saxl-isa-j m-is Saul-is-isa-j (Old Georgian: Boeder'95)
  all-N the-N blood-N house-G-N the-N Saul-G-G-N
  'all the blood of the house of Saul'

nb: mirror! but cf. Koopman'05, Evans'95, et al for anti-mirror, etc



Approximate Distribution of Aufnahme Languages

map from Plank'95

…many more if **hidden stacking** explains 'default case' etc
(Moravcsik'95;Svenonius'05;Richards'07;Matushansky'11;Bratttico'11)

Pesetsky'10:

èt-i posledn-ie karsiv-ye stol-y
these-N.PL last-N.PL beautiful-N.PL table-N.PL

èt-i posledn-ie dva karsiv-yx stol-a
these-N.PL last-N.PL two-N beautiful-G.PL table-G.SG

(16) **Case marking/concord parameters of variation**

- **what cases spell out:** local only;… all stacked
- **where cases spell out:** head; phrase edge; … complete
- **how cases spell out:** mirror, anti-mirror,…

(17) **Thm:** stacked+complete case marking
$[N_1\text{-}K_1[N_2\text{-}K_2\text{-}K_1[N_3\text{-}K_3\text{-}K_2\text{-}K_1\ldots[N_n\text{-}K_n\text{-}\ldots\text{-}K_1]]]]$
⇒ not MCS (Michaelis&Kracht'96;Kracht'02)

(18) **Does stacked+complete marking ever happen?**

- Kayardild? (Round'10,Evans'95 vs M&K)
- Old Georgian? (M&K vs Bhatt&Joshi'04)

…evidence not robust, but it is there

(19) **How to get case marking/concord and variability?** 3 options in literature:

(O1) merge, with complex feature 'copying', 'percolation'
with variation in either (i) what is copied, or (ii) what is spelled out
(apparently cannot work without complex feature structures – vs. Sportiche'98, Adger'10)

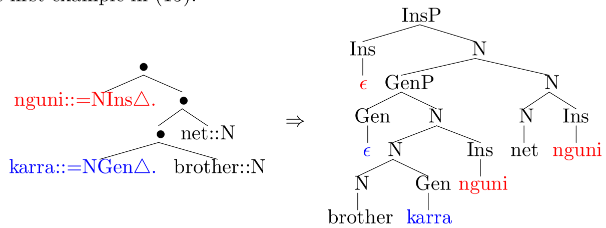
(O2) new syntactic operations 'multiple agree', 'polyvalent case', etc.
(not head-head or head-phrase; requires infinite categorization;
not 'minimal requirement for intelligibility')

(O3) case concord in spellout (with head movement, agreement, etc)

(20) **(O3) formal model: MGSO.** $Lex \subseteq Ph \times F^* \times$ Mode, where Mode specifies spellout

$\triangledown$. r spellout on r edge of overt heads in complement
□. r spellout on r edge of complement
△. r spellout on r edge of all overt heads in complement
… more (see appendix for superset proposal)

E.g. to derive the first example in (15):



- This takes us beyond MCS. ES conjectures these languages still 'approx MCS', PMCF.

At the interface: Chinese number names

(21) a. six million five thousand four
b. *five thousand six million four

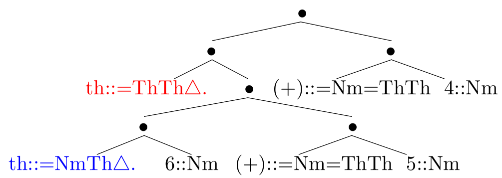
Suppose we did not have 'million' or 'billion', that 'thousand' was the largest number word. Then the language would have a largest number name unless it had some sort of recursive rule, like using 'thousand thousand' for 'million' and so on. Chinese has something like this (though the actual details of speakers' knowledge of these names is exceedingly complex, as it is for English speakers'!).

(22) a. six thousand thousand, five thousand four
b. *five thousand, six thousand thousand four

(23) $[[[N_1\text{-}Th^{n_1}]N_2\text{-}Th^{n_2}]\ldots N_n\text{-}Th^{n_n}]$ where $n_1 > n_2 > \ldots > n_n$

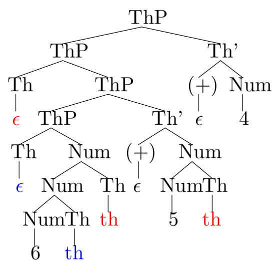
Notice that is essentially the same pattern as complete case marking.

(24) **Thm** This pattern is not MCS, but easily generated by MGSO (Radzinski'91)



in other words: $(1000\times((1000\times 6)+5))+4$

nb: merges shown here in Spec-Head-Comp order, unlike the literature



Contrast Ionin&Matushansky'06, etc.

Timing proposals compared

Richards'07 on case and 'future spreading' in Lardil:

(25) a. Ngada ji-jarri karnjin-i
I eat-NEG wallaby-ACC
'I didnt eat the wallaby'

b. Ngada ji-neng-kur karnjin-kur
I eat-NEG-FUT wallaby-FUT
'I won't eat the wallaby'

. . . the direct objects of both sentences . . . are first assigned Accusative case by v, and Accusative case is then deleted, again by v. In [(25b)], the T head later assigns Future morphology to the direct object.

Brattico'11 on Finnish:

(26) a. [yksi mies] nukkui
one-nom man-NOM slept
'one man slept'

b. minä näin [yhde-n miehe-n]
I saw one-ACC man-ACC
'I saw one man'

(27) a. [kaksi mies-tä] nukkui
two man-SG.PRT slept
'two men slept'

b. minä näin [kaksi mies-tä]
I saw two man-SG.PRT
'I saw two men'

Multi-agreement makes it possible to have one probe assigning case to several goals . . . but it does not allow several probes to affect the one and the same goal. . . The trick is that we do not limit Agree by phase boundaries (no phase/PIC condition), we do not prevent probes to get goals which already have case features (no activity condition), and we allow probes to search over goals (no intervention). . . . Case concord is subject to a range of possibilities that virtually exhaust the space of logical possibilities. . .

Pesetsky'10 on Russian:

(28) a. èt-i posledn-ie karsiv-ye stol-y
these-N.PL last-N.PL beautiful-N.PL table-N.PL

b. èt-i posledn-ie dva karsiv-yx stol-a
these-N.PL last-N.PL two-N beautiful-G.PL table-G.SG

Pesetsky proposes:

Feature Assignment (FA), version 3 of 5

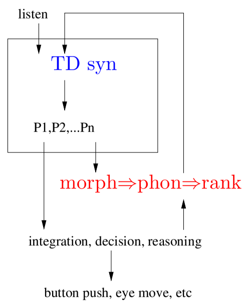
a. Copying: When α merges with β, forming $[_{\alpha}\alpha\ \beta]$, if α has satisfied its complementation requirements and is designated as a feature-assigner for β, its grammatical features are immediately copied onto β...

b. Realization: and are realized as morphology on all lexical items dominated by β.

Notice 2 things about a. First, if higher cases are to be distinguished from lower ones (in order to spell them out in the right order), the "copying" must actually build some structure (structure that is recursive, if stacking is recursive). Second, the only reason for copying these features is to get that spellout right, so why put this into the syntax at all? Also notice about b: this is the obvious spellout rule △. that we have used above (also the one of Kracht'02 and earlier work too). (His version 5 of 5 has these features but introduces some additional complexities which we do not need to get into here.)

Tentative conclusions, work in progress

- modular structure in G: diff kinds of operations
- transparent realization in performance: diff kinds of operations



Syn:
- Syn composition, recursion is simple (RTG)
- '2 step' models: LF ← Syn → PF
- Simplest parsers discovered recently

P, stress, case, chinese numbers
- depend on unbounded structure
- ¬'expression of featural makeup of the head'
- ¬'feature copying' or 'percolation'

Syn→PF takes us to 'approx MCS'

- timing claims = modularity claims
- todo:
  - define range of SO functions for attested case patterns (including structural/inherent case interactions, differential case marking, interaction with agreement, etc), and assess conjecture: MGSO⊂PMCF
  - formulate the other options of (19) with expected result: O1≡O2≡O3
  - extend to hm and compare other options: hm as merge; hm as reprojection; etc

    In the 2-step 'Derive+SO' models (cf page 1), it is natural to consider variations in SO, but if we allow anything like the SO functions defined here, this powerful mechanism alone can explain all the head movement and agreement effects. (cf. Bobaljik and others for similar proposals). The bearing of these effects on syntactic structure becomes less direct. A report on this in preparation: Stabler'11c
  - embed in performance model (immediate!) and assess

References

[1] ADGER, D. A minimalist theory of feature structure. In *Features: Perspectives on a key notion in linguistics*, A. Kibort and G. Corbett, Eds. Oxford University Press, Oxford, 2010, pp. 185–218.

[2] ANDERSON, S. R. *On Case Grammar.* Humanities Press, Atlantic Highlands, New Jersey, 1977.

[3] ANDERSON, S. R. Where's morphology. *Linguistic Inquiry 13* (1982), 571–612.

[4] ANDERSON, S. R. *A-Morphous Morphology.* Cambridge University Press, NY, 1992.

[5] ANDREWS, A. D. Semantic case-stacking and inside-out unification. *Australian Journal of Linguistics 15* (1996), 1–55.

[6] BECKER, T., JOSHI, A. K., AND RAMBOW, O. Long-distance scrambling and tree adjoining grammars. In *Proceedings of the Fifth Conference of the European Association for Computational Linguistics* (Morristown, New Jersey, 1991), Association for Computational Linguistics, pp. 21–26.

[7] BECKER, T., RAMBOW, O., AND NIV, M. The derivational generative power of formal systems, or, scrambling is beyond LCFRS. IRCS technical report 92-38, University of Pennsylvania, 1992.

[8] BHATT, R., AND JOSHI, A. Semilinearity is a syntactic invariant: a reply to Michaelis and Kracht. *Linguistic Inquiry 35* (2004), 683–692.

[9] BOBALJIK, J. D. Where's ϕ? Agreement as a post-syntactic operation. In *Leiden Papers in Linguistics*, M. van Koppen, Ed. University of Leiden, Leiden, 2005.

[10] BOECKX, C., AND STJEPANOVIC, S. Head-ing toward PF. *Linguistic Inquiry 32* (2001), 345–355.

[11] BOEDER, W. Suffixaufnahme in Kartvelian. In *Double Case: Agreement by Suffixaufnahme*, F. Plank, Ed. Oxford University Press, NY, 1995.

[12] BRATTICO, P. Case assignment, case concord, and the quantificational case construction. *Lingua 121*, 6 (2011), 1042–1066.

[13] CHO, T., AND KEATING, P. Articulatory strengthening at the onset of prosodic domains in korean. *Journal of Phonetics 28* (2001), 155–190.

[14] CHOMSKY, N. *The Minimalist Program.* MIT Press, Cambridge, Massachusetts, 1995.

[15] CHOMSKY, N. Derivation by phase. MIT. Forthcoming, 1999.

[16] CHOMSKY, N. Minimalist inquiries: The framework. In *Step by Step: Essays on Minimalism in Honor of Howard Lasnik*, R. Martin, D. Michaels, and J. Uriagereka, Eds. MIT Press, Cambridge, Massachusetts, 2000, pp. 89–155.

[17] DENCH, A., AND EVANS, N. Multiple case-marking in Australian languages. *Australian Journal of Linguistics 8* (1988), 1–47.

[18] EVANS, N. *A grammar of Kayardild.* Mouton de Gruyter, Dordrecht, 1995.

[19] EVANS, N. Multiple case in Kayardild: Anti-iconic suffix ordering and the diachronic filter. In *Double case: Agreement by Suffixaufnahme*, F. Plank, Ed. Oxford University Press, NY, 1995, pp. 396–430.

[20] FOUGERON, C., AND KEATING, P. A. Articulatory strengthening at edges of prosodic domains. *Journal of the Acoustic Society of America 101* (1997), 3728–3740.

[21] FRANK, R., AND SATTA, G. Optimality theory and the generative complexity of constraint violability. *Computational Linguistics 24* (1998), 307–315.

[22] FRAZIER, L. *On Comprehending Sentences: Syntactic Parsing Strategies.* PhD thesis, University of Massachusetts, Amherst, 1978.

[23] GRAF, T. Closure properties of minimalist derivation tree languages. In *Logical Aspects of Computational Linguistics, LACL'11* (2011).

[24] GUILLAUMIN, M. Conversions between mildly sensitive grammars. UCLA and École Normale Supérieure. http://www.linguistics.ucla.edu/people/stabler/epssw.htm, 2004.

[25] HARKEMA, H. A characterization of minimalist languages. In *Logical Aspects of Computational Linguistics* (NY, 2001), P. de Groote, G. Morrill, and C. Retoré, Eds., Lecture Notes in Artificial Intelligence, No. 2099, Springer, pp. 193–211.

[26] HEINZ, J. Learning long-distance phonotactics. *Linguistic Inquiry 41*, 4 (2010), 623–661.

[27] IONIN, T., AND MATUSHANSKY, O. The composition of complex cardinals. *Journal of Semantics 23* (1992), 315–360.

[28] JÄGER, G. Gradient constraints in finite state OT: The unidirectional and the bidirectional case. In *More than Words: A Festschrift for Dieter Wunderlich*, J. Greenberg, Ed. Akademie Verlag, Berlin, 2002, pp. 299–325.

[29] JOSHI, A. How much context-sensitivity is necessary for characterizing structural descriptions. In *Natural Language Processing: Theoretical, Computational and Psychological Perspectives*, D. Dowty, L. Karttunen, and A. Zwicky, Eds. Cambridge University Press, NY, 1985, pp. 206–250.

[30] KEATING, P., CHO, T., FOUGERON, C., AND HSU, C.-S. Domain-initial articulatory strengthening in four languages. In *Phonetic Interpretation (Papers in Laboratory Phonology 6)*, R. T. J. Local, R. Ogden, Ed. Cambridge University Press, NY, 2003, pp. 143–161.

[31] KING, J., AND JUST, M. A. Individual differences in syntactic processing: the role of working memory. *Journal of Memory and Language 30* (1991), 580–602.

[32] KOBELE, G. M. *Generating Copies: An Investigation into Structural Identity in Language and Grammar.* PhD thesis, UCLA, 2006.

[33] KOBELE, G. M. Minimalist tree languages are closed under intersection with recognizable tree languages. In *Logical Aspects of Computational Linguistics, LACL'11* (2011).

[34] KOBELE, G. M., RETORÉ, C., AND SALVATI, S. An automata-theoretic approach to minimalism. In *Model Theoretic Syntax at 10, ESSLLI'07* (2007), J. Rogers and S. Kepser, Eds.

[35] KOOPMAN, H. Korean (and Japanese) morphology from a syntactic perspective. *Linguistic Inquiry 36*, 4 (2005), 601–633.

[36] Kracht, M. Combinatorics of cases. *Research in Language and Computation 1*, 1 (2002), 59–97.

[37] Mainguy, T. A probabilistic top-down parser for minimalist grammars. http://arxiv.org/abs/1010.1826v1, 2010.

[38] Merchant, J. Polyvalent case, geometric hierarchies, and split ergativity. In *Proceedings of the 42nd annual meeting of the Chicago Linguistics Society, Volume 2: The Parasessions*, J. Bunting, S. Desai, R. Peachey, C. Straughn, and Z. Tomkova, Eds. Chicago Linguistic Society, Chicago, 2006, pp. 47–67.

[39] Michaelis, J. Derivational minimalism is mildly context-sensitive. In *Proceedings, Logical Aspects of Computational Linguistics, LACL'98* (NY, 1998), Springer, pp. 179–198.

[40] Michaelis, J. Transforming linear context free rewriting systems into minimalist grammars. In *Logical Aspects of Computational Linguistics* (NY, 2001), P. de Groote, G. Morrill, and C. Retoré, Eds., Lecture Notes in Artificial Intelligence, No. 2099, Springer, pp. 228–244.

[41] Michaelis, J. Notes on the complexity of complex heads in a minimalist grammar. In *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks, TAG+6* (2002), pp. 57–65.

[42] Michaelis, J., and Kracht, M. Semilinearity as a syntactic invariant. In *Logical Aspects of Computational Linguistics* (NY, 1997), C. Retoré, Ed., Springer-Verlag (Lecture Notes in Computer Science 1328), pp. 37–40.

[43] Michaelis, J., Mönnich, U., and Morawietz, F. Algebraic description of derivational minimalism. In *International Conference on Algebraic Methods in Language Processing, AMiLP'2000/TWLT16* (2000).

[44] Michaelis, J., Mönnich, U., and Morawietz, F. On minimalist attribute grammars and macro tree transducers. In *Linguistic Form and its Computation*, C. Rohrer, A. Rossdeutscher, and H. Kamp, Eds. CSLI Publications, Stanford, California, 2001, pp. 287–326.

[45] Mönnich, U. TAGs M-constructed. In *TAG+ Workshop, Institute for Research in Cognitive Science, University of Pennsylvania* (1998).

[46] Mönnich, U. Minimalist syntax, multiple regular tree grammars, and direction preserving tree transductions. In *Model Theoretic Syntax at 10. ESSLLI'07 Workshop Proceedings* (2007), J. Rogers and S. Kepser, Eds.

[47] Mönnich, U. Well-nested tree languages and attributed tree transducers. In *The 10th International Conference on Tree Adjoining Grammars and Related Formalisms TAG+10* (2010).

[48] Moravcsik, E. A. Summing up Suffixaufnahme. In *Double Case: Agreement by Suffixaufnahme*, F. Plank, Ed. Oxford University Press, NY, 1995.

[49] Morawietz, F. *Two Step Approaches to Natural Language Formalisms.* PhD thesis, University of Tübingen, 2001.

[50] Pesetsky, D. Russian case morphology and the syntactic categories. MIT.

[51] Plank, F. Re-introducing Suffixaufnahme. In *Double Case: Agreement by Suffixaufnahme*, F. Plank, Ed. Oxford University Press, NY, 1995.

[52] Pollard, C. *Generalized phrase structure grammars, head grammars and natural language.* PhD thesis, Stanford University, 1984.

[53] Radzinski, D. Chinese number names, tree adjoining languages and mild context sensitivity. *Computational Linguistics 17* (1991), 277–300.

[54] Rambow, O. *Formal and Computational Aspects of Natural Language Syntax.* PhD thesis, University of Pennsylvania, 1994. Computer and Information Science Technical report MS-CIS-94-52 (LINC LAB 278).

[55] Richards, N. Lardil case stacking and the structural/inherent case distinction. http://ling.auf.net/LingBuzz/000405, 2007.

[56] Roark, B., and Johnson, M. Efficient probabilistic top-down and left-corner parsing. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics* (1999), pp. 421–428.

[57] Round, E. R. *Kayardild Morphology, Phonology and Morphosyntax.* PhD thesis, Yale University, 2010.

[58] Schütze, C. T. On the nature of default case. *Syntax 4*, 3 (2001), 205–238.

[59] Seki, H., Matsumura, T., Fujii, M., and Kasami, T. On multiple context-free grammars. *Theoretical Computer Science 88* (1991), 191–229.

[60] Shieber, S. M. Unifying synchronous tree-adjoining grammars and tree transducers via bimorphisms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)* (2006).

[61] Stabler, E. P. Computational perspectives on minimalism. In *Oxford Handbook of Minimalism*, C. Boeckx, Ed. Oxford University Press, Oxford, 2010, pp. 617–641.

[62] Stabler, E. P. After GB theory. In *Handbook of Logic and Language, Second Edition*, J. van Benthem and A. ter Meulen, Eds. Elsevier, Amsterdam, 2011, pp. 395–414.

[63] Stabler, E. P. Three more approaches to head movement. UCLA, manuscript in preparation, 2011.

[64] Stabler, E. P. Top-down recognizers for MCFGs and MGs. In *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics (CMCL), 49th Annual Meeting of the Association for Computational Linguistics* (2011), F. Keller and D. Reitter, Eds.

[65] Stabler, E. P., and Keenan, E. L. Structural similarity. *Theoretical Computer Science 293* (2003), 345–363.

[66] Vijay-Shanker, K., Weir, D., and Joshi, A. Characterizing structural descriptions produced by various grammatical formalisms. In *Proceedings of the 25th Annual Meeting of the Association for Computational Linguistics* (1987), pp. 104–111.

[67] Villemonte de la Clergerie, E. Parsing MCS languages with thread automata. In *Proceedings of the 6th International Workshop on Tree Adjoining Grammars and Related Frameworks, TAG+6* (2002).

[68] Wagner, M. *Prosody and Recursion.* PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2005.

[69] Wagner, M. Prosody and recursion in coordinate structures and beyond. *Working Papers of the Cornell Phonetics Laboratory 16* (2007).

A. Defining SO modes with tree transducers

Realizing SO with TD tree transducers, they can traverse the partial tree completed by a TD parse, in parallel, defining the leaves for the check against what is heard (in CS this check against the input is called the 'scan' operation). This allows an extremely fast and transparent model, at least for simple spellout functions like the ones in the literature.

It is not yet clear how broad the class of spell-out functions must be for descriptive adequacy, but all the proposals I have seen are extremely easy to implement with a 1-state, deterministic, multiple extended TD tree transducer (with copying). This allows us to

- stack affixes to (left or right), or replace (so only most local case marked)
- attach to: (left or right) of (head or phrase edge or all overt heads)

The common mirror orders are left-stacking left-attaching and right-stacking right-attaching, which can be indicated by these notations

.	normal spellout
$\triangledown.$	r spellout on r edge of overt head of complement
$\Box.$	r spellout on r edge of complement
$\triangle.$	r spellout on r edge of all overt heads in complement
$.\triangledown$	l spellout on l edge of overt head of complement
$.\Box$	l spellout on l edge of complement
$.\triangle$	l spellout on l edge of all overt heads in complement

But of course it is easy to define 'antimirror' and other 'mixed' orders too. Certainly we do not need the whole space of deterministic, copying multiple TD tree transductions.

For example, consider the mode $\triangle.$ used in the examples above, in a grammar where this is the only non-standard spellout mode. As usual in MCF representations of MGs, categories are regarded as pairs $\langle T, F\rangle$ where $T \in \{0,1\}$ is a truth value signifying whether the category is lexical and F is a sequence of syntactic features. Terminal categories $\langle T, F, M\rangle$ where $T = 1$ and M is the spellout mode. Then we can compute the spellout function with a 1-state TD tree transducer which takes as an argument the tree it is processing and tree of elements to be right adjoined to the leaves (shown here simply as a sequence s or sw). This transducer collects in its tree argument the sequence to be adjoined in the complement:

$$q_0(\langle 0,\alpha\rangle(\langle 1,{=}f\alpha,\triangle.\rangle(w),\langle 0,f\rangle(x)),s) \rightarrow \langle 0,\alpha\rangle(q_0(\langle 1,{=}f\alpha\rangle(\epsilon),\epsilon),q_0(\langle 0,f\rangle(x),sw))$$

This rule shows 'right stacking'; for left stacking we simply build ws instead of sw. When processing a head-complement structure that is spelled out in the usual way, the adjoining sequence is passed through, unchanged:

$$q_0(\langle 0,\alpha\rangle(\langle 1,{=}f\alpha,.\rangle(w),\langle 0,f\rangle(x)),s) \rightarrow \langle 0,\alpha\rangle(q_0(\langle 1,{=}f\alpha\rangle(w),s),q_0(\langle 0,f\rangle(x),s))$$

At the leaves, the adjoining sequence is spelled out:

$$q_0(\langle 1,\alpha,w\rangle(),s) \rightarrow \alpha(w,s)$$

When s is empty, scan will simply check for w. When s is non-empty, scan will check for each element. This rule shows suffixing, 'right adjoining'; for prefixing we simply build $\alpha(s,w)$ instead of $\alpha(w,s)$.