Past Tense Marking in Afrikaans

Manfred Sailer
Sonderforschungsbereich 441, Universität Tübingen
mf@sfs.uni-tuebingen.de

Sinn und Bedeutung VIII
Johann Wolfgang Goethe-Universität
Frankfurt am Main
29.9.–1.10.2003

1 Introduction

Major claims:

- Afrikaans provides evidence for underspecification and multiple exponents of temporal operators.
- This constitutes empirical motivation for *Lexical Resource Semantics* as an appropriate framework for compositional semantics.

(1) Jan wou die boek gekoop het.
Jan wanted.IMP the book bought.PART AUX

a. Jan wanted to have bought the book.
PAST(Jan wants ˆPAST(Jan buys the book))

b. Jan wants to have bought the book.
Jan wants ˆPAST(Jan buys the book)

c. Jan wanted to buy the book.
PAST(Jan wants ˆ(Jan buys the book))

2 Data

2.1 Afrikaans Verbal Morphology

(2) Afrikaans tense forms:

	be	have	write	can do
infinitive	wees	hê	skryf	kan doen
presens	is	het	skryf	kan doen
imperfek	was	(had)	—	kon doen
perfek	het/was gewees	het gehad	het geskryf	(het kan doen)
presens-perfek	—	—	—	kan gedoen het
imperfek-perfek	—	—	—	kon gedoen het
perfek-imperfek	—	—	—	(het kon doen)

2.2 The Interpretation of Tenses in Simple Clauses

(3) a. Hy het dadelik huis toe gestap.
he AUX really house towards stepped.PART 'He really stepped towards his house.'

b. Toe stap hy dadelik huis toe.
Then goes he really house towards 'Then, he really stepped towards his house.'

c. Verlede week stap hy huis toe, en daar sien hy sy buurman voor die hekkie.
last week steps he house towards and there sees he his neighbor in front of the hedge
'Last week, he stepped towards his house and there he saw his neighbor in front of the hedge.'

The following observations hold for regular verbs:

- If there is no other indication for anteriority, the *perfek* is used.
- If there is another indication for anteriority, the *presens* can be used.

For verbs which have an *imperfek* form, there is a tendency to use this *imperfek* form in all simple past contexts.

(4) a. Hy was dadelik tuis.
he was.IMP really home

b. Toe was/ is hy dadelik tuis.
then was.IMP/ is.PRES he really home

c. Verlede week was/ is hy dadelik tuis en daar kom sy maat op besoek.
last week was.IMP/ is.PRES he really home and his friend comes to see him.

(5) a. Hy wou huis toe stap.
he wanted.IMP house towards step

b. Toe wou/ wil hy huis toe stap.
then wanted.IMP/ wants.PRES he house towards step

c. Verlede week wou/ wil hy huis toe stap en daar sien hy sy buurman ...
last week wanted.IMP/ wants.PRES he house towards step ...

2.2.1 *Perfek* as Past

(6) a. Môre sien ek hom.
tomorrow see.PRES I him 'I'll see him tomorrow.'

b. * Môre het ek hom gesien (en dan sal ek alles vir jou vertel).
tomorrow AUX I him seen.PART and then will I everything to you tell

c. Môre sal ek hom gesien het ...
tomorrow will I him seen.PART AUX ...

(7) a. Jan het gebel.
Jan AUX call.PART 'Jan called.'

b. PAST($\mathsf{call}'(j)$)

c. $[\![\text{PAST}(\phi)]\!]^{s,g,M} = 1$ iff there is a time t before s such that $[\![\phi]\!]^{t,g,M} = 1$.

2.2.2 *Imperfek* as Past

(8) a. Môre is Jan tuis.
tomorrow is.PRES Jan home 'Jan will be home tomorrow.'

b. * Môre was Jan tuis.
tomorrow was.IMP Jan home

c. Môre sal Jan tuis gewees het.
tomorrow will Jan home been.PART AUX 'Jan will have been home tomorrow.'

(9) a. Jan was tuis.
Jan was.IMP home 'Jan was home.'

b. PAST($\mathsf{be\text{-}home}'(j)$)

2.3 The Interpretation of Tense in Verbal Complexes

The scope of PAST is not fully determined by its syntactic position:

(10) Ek moet los kon rondgeloop het.
I must.PRES freely can.IMP around.walked.PART AUX

'I had to be able to run around freely.' (Kleij, 1999)
PAST(must′(^can′(i, ^run-around-freely′(i))))

Generalization 1: The scope of PAST is clause-bound, but not fully determined by the verb that introduces the operator.

If there is a *perfek* or an *imperfek*, then there is a PAST operator in the logical form of the sentence, a present interpretation is not possible.

(11) a. Jan het die boek gelees.
Jan AUX the book read.PART

PAST: 'Jan read the book.'
– \$ 'Jan is reading the book.'

b. Jan wou die boek lees.
Jan wanted.IMP the book read

PAST > want′: 'Jan wanted to read the book.'
want′ > PAST: 'Jan wants to have read the book.'
want′: \$ 'Jan wants to read the book.'

There cannot be more PAST operators in the logical form than there are *perfek* or *imperfek* verb forms:

(12) a. Jan het die boek gelees.
Jan AUX the book read.PART

PAST: 'Jan read the book.'
PAST > PAST: \$ 'Jan had read the book.'

b. Jan wou die book lees.
Jan wanted.AUX the book read

PAST > want′: 'Jan wanted to read the book.'
want′ > PAST: 'Jan wants to have read the book.'
PAST > want′ > PAST: \$ 'Jan wanted to have read the book.'

Generalization 2: Verbs in *perfek* or *imperfek* contribute a PAST operator to the logical form.

(13) Jan wou die boek gekoop het.
Jan wanted.IMP the book bought.PART AUX

a. Jan wanted to have bought the book.
PAST(want′(j, ^PAST(buy′(j, the-book))))

b. Jan wants to have bought the book.
want′(j, ^PAST(buy′(j, the-book)))

c. Jan wanted to buy the book.
PAST(want′(j, ^buy′(j, the-book)))

Generalization 3: The number of *perfek* and *imperfek* verbs determines the upper-bound of the number of PAST operators in a clause, but not the exact number.

3 Lexical Resource Semantics

General remarks:

- LRS uses techniques of *underspecified semantics* (Reyle, 1993; Bos, 1996), but the logical form of a sentence is a single, disambiguated expression of the semantic representation language.
- Previous LRS publications discuss scope ambiguity (Richter and Sailer, 2001a; Bouma, 2003), negative concord (Richter and Sailer, 2001b) and multiple wh-interrogatives in German (Richter and Sailer, 2001a).
- Richter and Sailer (2003) provides an introduction to LRS.
- Previous work on LRS was closely linked to the framework of Head-Driven Phrase Structure Grammar (HPSG, Pollard and Sag (1994)). The present presentation is rather neutral with respect to a particular framework. In Sailer (2003) a more HPSG-oriented version of the present analysis can be found.
- The HPSG-independent notation of LRS is joint work with Gerald Penn and Frank Richter.

3.1 An Outline of LRS

(14) The semantic meta language $\mu e(L)$:

For a semantic representation language L, the set of *meta expressions* of L ($\mu e(L)$) is the smallest set such that,

there is a set VAR of meta variables, and
for each $\phi \in L$, $\phi \in \mu e(L)$,
for each $\alpha \in VAR$, for each n-tuple $\phi_1, \dots \phi_n$ with $\phi_i \in \mu e(L)$, $\alpha[\phi_1, \dots . \phi_n] \in \mu e(L)$,
for each $\phi, \psi \in \mu e(L)$, for each variable v of L, $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, …, $\exists v\phi$, $\forall v\phi \in \mu e(L)$.

We write $\vec{\phi}$ for an n-tuple of expressions from $\mu e(L)$. We also allow 0-tuples, where we abbreviate $\alpha[]$ as α.

(15) The denotation of meta expressions is defined as follows:
Let $A : VAR \mapsto L$ be a meta variable assignment, then

for each $\phi \in L$, $[\![\phi]\!]^A = \phi$,
for each $n \in \mathbb{N}$, for each $\alpha \in VAR$, for each n-tuple $\phi_1, \dots \phi_n$ with $\phi_i \in \mu e(L)$,
$[\![\alpha[\phi_1, \dots . \phi_n]]\!]^A = A(\alpha)$, where for each $i \leq n$, $[\![\phi_i]\!]^A$ is a subexpression of $A(\alpha)$.
for each $\phi, \psi \in \mu e(L)$, for each variable v of L,
$[\![\neg\phi]\!]^A = \neg[\![\phi]\!]^A$
$[\![\phi \wedge \psi]\!]^A = [\![\phi]\!]^A \wedge [\![\psi]\!]^A$
$\vdots$
$[\![\forall v\phi]\!]^A = \forall v[\![\phi]\!]^A$

(16) Reading
For each $\phi \in \mu e(L)$, $\phi_{lf} \in L$, ϕ_{lf} *is a reading* of ϕ, iff there is a meta variable assignment function A such that

$\phi_{lf} = [\![\phi]\!]^A$, and
for each $\alpha \in VAR$ that is a subexpression of ϕ,
$A(\alpha)$ is a subexpression of ϕ, or
there are $\beta_1, \dots, \beta_n \in VAR$, and $\vec{\psi_1}, \dots, \vec{\psi_n} \in \mu e(L)$ such that
$A(\alpha)[\beta_1[\vec{\psi_1}]/A(\beta_1), \dots, \beta_n[\vec{\psi_n}]/A(\beta_n)]$ is a subexpression of ϕ.

(17) Definition: *lrs*
An *lrs* is a triple $\langle\phi, \phi_-, \phi_\#\rangle$ of $\mu e(L)$ (called: the *parts* of the *lrs*, the *internal content* and the *external content* of the *lrs*), such that ϕ_- is a subexpression of ϕ and there is an A such that $\phi_\#$ is a subexpression of $[\![\phi]\!]^A$.
A *constraint lrs* is a pair $\langle\lambda, \kappa\rangle$, where λ is an *lrs* and κ is a set of constraints of the form

$\phi \lhd \alpha$, where $\phi \in \mu e(L)$, and $\alpha \in VAR$ both occurs in λ, or
$\phi = \psi$, where ϕ, ψ both occur in λ.

.

(18) Constraint Elimination

Let $\langle \lambda, \kappa \rangle$ be a constraint lrs. Each constraint $c \in \kappa$, c can be eliminated by the following rule:

if c is of the form $\phi \lhd \alpha$, then

for each occurrence of α in λ such that there is a $\vec{\psi} \in \mu e(L)$ which is of the form $\alpha[\vec{\psi}]$.

replace $\alpha[\vec{\psi}]$ with $\alpha[\vec{\psi}, \phi]$

if c is of the form $\phi = \psi$, then

let $\alpha \in VAR$ be a meta variable that does not occur in λ and replace each occurrence of ϕ or ψ in λ with $\alpha[\phi, \psi]$,

(19) Short-hand for lrs:

parts (ϕ)	lrs incont (ϕ_-)	excont ($\phi_\#$)	short hand
$\langle \alpha[\mathsf{call'}(j)]$,	$\mathsf{call'}(j)$,	$\alpha[\mathsf{call'}(j)]\rangle$,	${}^\#\alpha[\underline{\mathsf{call'}(j)}]$
$\langle \neg\alpha[\exists x[\beta[\mathsf{human'}(x)] \wedge \gamma]]$,	$\mathsf{human'}(x)$,	$\exists x[\beta[\mathsf{human'}(x)] \wedge \gamma]\rangle$	$\neg\alpha[{}^\#\exists x[\beta[\underline{\mathsf{human'}(x)}] \wedge \gamma]]$

(20) Semantics Principle:

Let $\langle \phi, \phi_-, \phi_\# \rangle$ be the lrs of the head daughter,
$\langle \psi, \psi_-, \psi_\# \rangle$ the lrs of the nonhead daughter,
and α a meta variable that does not occur in either lrs,

then the lrs of the mother results from eliminating every constraint from the following constraint lrs:
$\langle\langle \alpha[\phi', \psi'], \phi_-, \phi_\# \rangle, \kappa\rangle$,

where κ contains exactly the following constraints:

1. $\phi_\#$ is of the form $\beta[\vec{\phi}]$, and $\psi_\# \lhd \beta$ is in κ,
2. Specific Constraints:
   (a) If the non-head is a quantified NP complement, then its $\psi_\#$ is of the form $Qx[\psi' \circ \beta]$, for some quantifier Q, some logical connector $\circ$, some variable x, some expression ψ' and some meta variable β, and $\phi_- \lhd \gamma$ is in κ.
   (b) If the non-head is a raised complement of the head, then $\phi_- = \psi_-$ and $\phi_\# = \psi_\#$.
   (c) ...

(21) The ExCont Principle:

Let $\langle \phi, \phi_-, \phi_\# \rangle$ be the lrs of an utterance, then $\phi_\#$ is a reading of ϕ.

Calculating with LRS

(22) Elimination of meta variables:

The relation $\rightsquigarrow \subseteq \mu e(L) \times \mu e(L)$ defines steps to eliminate meta variables as follows:

(R1) Variable Elimination: For each $\alpha \in VAR$, $\phi \in \mu e(L)$: $\alpha[\phi] \rightsquigarrow \phi$.

(R2) Formula Elimination: For each $\alpha \in VAR$, $\vec{\phi}, \psi \in \mu e(L)$: $\alpha[\vec{\phi}, \psi] \rightsquigarrow \alpha[\vec{\phi}]$ if there is a ϕ_i such that ψ is a subexpression of ϕ_i.

(R3) Subordination: For each $\alpha, \beta \in VAR$, $\vec{\phi}, \vec{\phi'}, \psi \in \mu e(L)$: $\alpha[\vec{\phi}, \psi] \rightsquigarrow \alpha[\vec{\phi}[\beta[\vec{\phi'}, \psi]/\beta[\vec{\phi'}]], \psi]$, if there is at least one occurrence of β in $\vec{\phi}$.

(23) Derivation of readings:

For each $\phi, \psi \in \mu e(L)$, $\phi \rightsquigarrow^* \psi$ iff one of the following holds

either $\phi = \psi$,
or there are $\phi', \psi' \in \mu e(L)$ such that

ϕ' is a subexpression of ϕ, and $\phi' \rightsquigarrow \psi'$, and $\phi[\psi'/\phi'] \rightsquigarrow^* \psi$

3.2 Examples

Simple example

(24) a. (dat) iemand bel
that someone calls 'that someone is calling'

b. *bel*: ${}^\#\alpha[\underline{\mathsf{call'}(j)}]$
iemand: ${}^\#\exists x[\underline{\mathsf{human'}(x)} \wedge \beta]$

c.
S
$\gamma[{}^\#\alpha[\underline{\mathsf{call'}(x)}], \exists x[\mathsf{human'}(x) \wedge \beta]]$
$\exists x[\mathsf{human'}(x) \wedge \beta] \lhd \alpha$
$\mathsf{call'}(x) \lhd \beta$
COMP HEAD
NP: ${}^\#\exists x[\underline{\mathsf{human'}(x)} \wedge \beta]$ *someone*
V: ${}^\#\alpha[\underline{\mathsf{call'}(x)}]$ *bel*

d. Constraint eliminated lrs:

$\gamma[{}^\#\alpha[\underline{\mathsf{call'}(x)}], \exists x[\mathsf{human'}(x) \wedge \beta]]$
$\gamma[{}^\#\alpha[\underline{\mathsf{call'}(x)}, \exists x[\mathsf{human'}(x) \wedge \beta]], \exists x[\mathsf{human'}(x) \wedge \beta]]$
$\gamma[{}^\#\alpha[\underline{\mathsf{call'}(x)}, \exists x[\mathsf{human'}(x) \wedge \beta[\underline{\mathsf{call'}(j)}]]], \exists x[\mathsf{human'}(x) \wedge \beta[\underline{\mathsf{call'}(j)}]]]$

e. Disambiguation:

$\gamma[\alpha[\mathsf{call'}(x), \exists x[\mathsf{human'}(x) \wedge \beta[\mathsf{call'}(j)]]], \exists x[\mathsf{human'}(x) \wedge \beta[\mathsf{call'}(j)]]]$
$\rightsquigarrow$(R2) $\gamma[\alpha[\mathsf{call'}(x), \exists x[\mathsf{human'}(x) \wedge \beta[\mathsf{call'}(j)]]]]$
$\rightsquigarrow$(R1) $\alpha[\mathsf{call'}(x), \exists x[\mathsf{human'}(x) \wedge \beta[\mathsf{call'}(j)]]]$
$\rightsquigarrow$(R2) $\alpha[\exists x[\mathsf{human'}(x) \wedge \beta[\mathsf{call'}(j)]]]$
$\rightsquigarrow$(R1, twice) $\exists x[\mathsf{human'}(x) \wedge \mathsf{call'}(j)]$

Scope ambiguity

(25) a. (dat) iedereen iets lees
(that) everyone something reads.PRES 'that everyone is reading something'

b.
${}^\#\alpha[\underline{\mathsf{read'}(x,y)}, \exists y[\mathsf{obj'}(y) \wedge \beta[\underline{\mathsf{read'}(x,y)}]], \forall x[\mathsf{hum'}(x) \rightarrow \gamma[\underline{\mathsf{read'}(x,y)}]]]$
S
$\epsilon[{}^\#\alpha[\underline{\mathsf{read'}(x,y)}, \exists y[\mathsf{obj'}(y) \wedge \beta[\underline{\mathsf{read'}(x,y)}]]], \forall x[\mathsf{hum'}(x) \rightarrow \gamma]]$
$\forall x \ldots \lhd \alpha$, $\mathsf{read'}(x,y) \lhd \gamma$
COMP HEAD
NP: ${}^\#\forall x[\underline{\mathsf{hum'}(x)} \rightarrow \gamma]$ *iedereen*
${}^\#\alpha[\underline{\mathsf{read'}(x,y)}, \exists y[\mathsf{obj'}(y) \wedge \beta[\underline{\mathsf{read'}(x,y)}]]]$
VP
$\delta[{}^\#\alpha[\underline{\mathsf{read'}(x,y)}], \exists y[\mathsf{obj'}(y) \wedge \beta]]$
$\exists y \ldots \lhd \alpha$, $\mathsf{read'}(x,y) \lhd \beta$
COMP HEAD
NP: ${}^\#\exists y[\underline{\mathsf{obj'}(y)} \wedge \beta]$ *iets*
V: ${}^\#\alpha[\underline{\mathsf{read'}(x,y)}]$ *lees*

c. Disambiguation:

- $\exists\forall$-reading:

$\alpha[\mathsf{read'}(x,y), \exists y[\mathsf{obj'}(y) \wedge \beta[\mathsf{read'}(x,y)]], \forall x[\mathsf{hum'}(x) \rightarrow \gamma[\mathsf{read'}(x,y)]]]$
$\rightsquigarrow$(R2) $\alpha[\exists y[\mathsf{obj'}(y) \wedge \beta[\mathsf{read'}(x,y)]], \forall x[\mathsf{hum'}(x) \rightarrow \gamma[\mathsf{read'}(x,y)]]]$
$\rightsquigarrow$(R3) $\alpha[\exists y[\mathsf{obj'}(y) \wedge \beta[\mathsf{read'}(x,y), \forall x[\mathsf{hum'}(x) \rightarrow \gamma[\mathsf{read'}(x,y)]]]], \forall x[\mathsf{hum'}(x) \rightarrow \gamma[\mathsf{read'}(x,y)]]]$
$\rightsquigarrow$(R2) $\alpha[\exists y[\mathsf{obj'}(y) \wedge \beta[\mathsf{read'}(x,y), \forall x[\mathsf{hum'}(x) \rightarrow \gamma[\mathsf{read'}(x,y)]]]]]$
$\rightsquigarrow$(R2) $\alpha[\exists y[\mathsf{obj'}(y) \wedge \beta[\forall x[\mathsf{hum'}(x) \rightarrow \gamma[\mathsf{read'}(x,y)]]]]]$
$\rightsquigarrow$(R1, three times) $\exists y[\mathsf{obj'}(y) \wedge \forall x[\mathsf{hum'}(x) \rightarrow \mathsf{read'}(x,y)]]$

- $\forall\exists$-reading:

  $\alpha[\mathsf{read}'(x,y), \exists y[\mathsf{obj}'(y) \wedge \beta[\mathsf{read}'(x,y)]], \forall x[\mathsf{hum}'(x) \rightarrow \gamma[\mathsf{read}'(x,y)]]]$
  $\leadsto$(R2) $\alpha[\exists y[\mathsf{obj}'(y) \wedge \beta[\mathsf{read}'(x,y)]], \forall x[\mathsf{hum}'(x) \rightarrow \gamma[\mathsf{read}'(x,y)]]]$
  $\leadsto$(R3) $\alpha[\exists y[\mathsf{obj}'(y) \wedge \beta[\mathsf{read}'(x,y)]], \forall x[\mathsf{hum}'(x) \rightarrow \gamma[\mathsf{read}'(x,y), \exists y[\mathsf{obj}'(y) \wedge \beta[\mathsf{read}'(x,y)]]]]]$
  $\leadsto$(R2 twice) $\alpha[\forall x[\mathsf{hum}'(x) \rightarrow \gamma[\exists y[\mathsf{obj}'(y) \wedge \beta[\mathsf{read}'(x,y)]]]]]$
  $\leadsto$(R1, three times) $\forall x[\mathsf{hum}'(x) \rightarrow \exists y[\mathsf{obj}'(y) \wedge \mathsf{read}'(x,y)]]$

het* and *wil

(26) a. (that) Jan gebel het
that Jan called.PART AUX '(that) Jan called'

b. *gebel*: $^{\#}\alpha[\underline{\mathsf{call}'(j)}]$
het: $^{\#}\delta[\textsc{past}(\beta[\underline{\gamma}])]$

c.
$^{\#}\delta[\textsc{past}(\beta[\underline{\mathsf{call}'(j)}])]$
S
$\zeta[^{\#}\delta[\textsc{past}(\beta[\underline{\mathsf{call}'(j)}]), \underline{\mathsf{call}'(j)}], j]$
COMP HEAD
NP $^{\#}\underline{j}$ *Jan*
$^{\#}\delta[\textsc{past}(\beta[\underline{\mathsf{call}'(j)}]), \underline{\mathsf{call}'(j)}]$
V
$\epsilon[^{\#}\delta[\textsc{past}(\beta[\underline{\gamma}])], \alpha[\mathsf{call}'(j)]]$
$\alpha[\ldots] \lhd \delta$
$\alpha = \delta,\ \gamma = \mathsf{call}'(j)$
COMP HEAD
V $^{\#}\alpha[\underline{\mathsf{call}'(j)}]$ *gebel*
V $^{\#}\delta[\textsc{past}(\beta[\underline{\gamma}])]$ *het*

d. Disambiguation:
$\delta[\textsc{past}(\beta[\mathsf{call}'(j)])]$
$\leadsto$(R1, twice) $\textsc{past}(\mathsf{call}'(j))$

(27) a. (dat) Jan wil bel
that Jan want.PRES call '(that) Jan wants to call'

b. *bel*: $^{\#}\alpha[\underline{\mathsf{call}'(j)}]$
wil: $^{\#}\beta[\mathsf{want}'(j, {}^\wedge\gamma[\underline{\delta}])]$

c.
$^{\#}\beta[\mathsf{want}'(j, {}^\wedge\gamma[\underline{\mathsf{call}'(j)}])]$
S
$\zeta[^{\#}\beta[\mathsf{want}'(j, {}^\wedge\gamma[\underline{\mathsf{call}'(j)}]), \underline{\mathsf{call}'(j)}], j]$
$j \lhd \beta$
COMP HEAD
NP $^{\#}\underline{j}$ *Jan*
$^{\#}\beta[\mathsf{want}'(j, {}^\wedge\gamma[\underline{\mathsf{call}'(j)}]), \underline{\mathsf{call}'(j)}]$
V
$\epsilon[^{\#}\beta[\mathsf{want}'(j, {}^\wedge\gamma[\underline{\delta}])], \alpha[\underline{\mathsf{call}'(j)}]]$
$\alpha[\ldots] \lhd \beta$
$\delta = \mathsf{call}'(j),\ \alpha = \beta$
HEAD COMP
V $^{\#}\beta[\mathsf{want}'(j, {}^\wedge\gamma[\underline{\delta}])]$ *wil*
V $^{\#}\alpha[\underline{\mathsf{call}'(j)}]$ *bel*

d. Disambiguation:
$\beta[\mathsf{want}'(j, {}^\wedge\gamma[\mathsf{call}'(j)])]$
$\leadsto$(R1, twice) $\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))$

4 Analysis

Generalization 1: The scope of PAST is underspecified:

(28) a. (dat) Jan wil gebel het
that Jan wants.PRES called.PART AUX

b. *gebel*: $^{\#}\alpha[\underline{\mathsf{call}'(j)}]$
het: $^{\#}\beta[\textsc{past}(\delta[\underline{\gamma}])]$
wil: $^{\#}\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\underline{\eta}])]$

c.
$^{\#}\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\underline{\mathsf{call}'(j)}]), \textsc{past}(\delta[\underline{\mathsf{call}'(j)}])]$
S
$\kappa[^{\#}\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\underline{\mathsf{call}'(j)}]), \textsc{past}(\delta[\underline{\mathsf{call}'(j)}])], j]$
$j \lhd \epsilon$
COMP HEAD
NP $^{\#}\underline{j}$ *Jan*
$^{\#}\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\underline{\mathsf{call}'(j)}]), \textsc{past}(\delta[\underline{\mathsf{call}'(j)}])]$
V
$\iota[^{\#}\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\underline{\eta}])], \beta[\textsc{past}(\delta[\mathsf{call}'(j)])]]$
$\beta[\ldots] \lhd \epsilon$
$\eta = \mathsf{call}'(j),\ \epsilon = \beta$
HEAD COMP
V $^{\#}\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\underline{\eta}])]$ *wil*
$^{\#}\beta[\textsc{past}(\delta[\underline{\mathsf{call}'(j)}])]$
V
$\theta[^{\#}\beta[\textsc{past}(\delta[\underline{\gamma}])], \alpha[\mathsf{call}'(j)]]$
$\alpha[\ldots] \lhd \beta$
$\mathsf{call}'(j) = \gamma,\ \alpha = \beta$
COMP HEAD
V $^{\#}\alpha[\underline{\mathsf{call}'(j)}]$ *gebel*
V $^{\#}\beta[\textsc{past}(\delta[\underline{\gamma}])]$ *het*

d. Disambiguation:

- Reading 1: $\mathsf{want}'(j, {}^\wedge\textsc{past}(\mathsf{call}'(j)))$

  $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)]), \textsc{past}(\delta[\mathsf{call}'(j)])]$
  $\leadsto$(R3) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j), \textsc{past}(\delta[\mathsf{call}'(j)])]), \textsc{past}(\delta[\mathsf{call}'(j)])]$
  $\leadsto$(R2, twice) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\textsc{past}(\delta[\mathsf{call}'(j)])])]$
  $\leadsto$(R1, three times) $\mathsf{want}'(j, {}^\wedge\textsc{past}(\mathsf{call}'(j)))$

- Reading 2: $\textsc{past}(\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j)))$

  $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)]), \textsc{past}(\delta[\mathsf{call}'(j)])]$
  $\leadsto$(R3) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)]), \textsc{past}(\delta[\mathsf{call}'(j), \mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)])])]$
  $\leadsto$(R2, twice) $\epsilon[\textsc{past}(\delta[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)])])]$
  $\leadsto$(R1, three times) $\textsc{past}(\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j)))$

Generalization 2: Verbs in *perfek* and *imperfek* contribute a PAST operator.

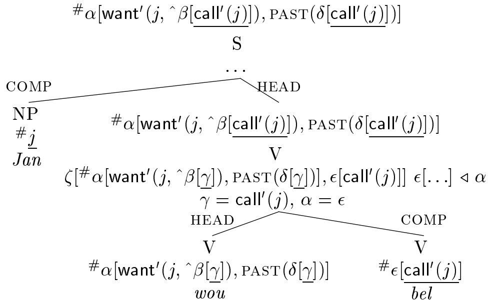
(29) a. *het*: $^{\#}\alpha[\textsc{past}(\beta[\underline{\gamma}])]$

b. *wou*: $^{\#}\alpha[\mathsf{want}'(j, {}^\wedge\beta[\underline{\gamma}]), \textsc{past}(\delta[\underline{\gamma}])]$

(30) a. (dat) Jan wou bel
that Jan wanted.IMP call

b. *bel*: $^{\#}\epsilon[\underline{\mathsf{call}'(j)}]$
wou: $^{\#}\alpha[\mathsf{want}'(j, {}^\wedge\beta[\underline{\gamma}]), \textsc{past}(\delta[\underline{\gamma}])]$

c.



d. Disambiguation:

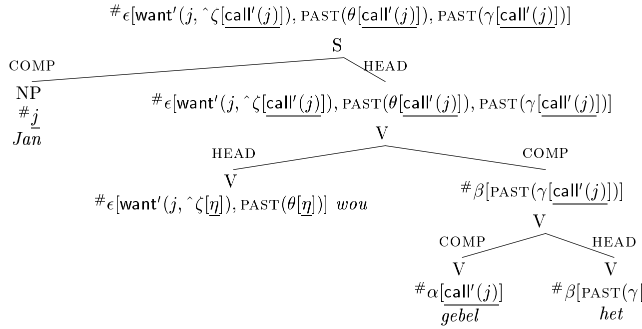
for the derivation of the readings see (28d).

Generalization 3: Multiple exponents.

(31) a. (dat) Jan wou gebel het
that Jan wanted.IMP called.PART AUX

b. *gebel*: $^{\#}\alpha[\underline{\mathsf{call}'(j)}]$
het: $^{\#}\beta[\textsc{past}(\gamma[\underline{\delta}])]$
wou: $^{\#}\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\underline{\eta}]), \textsc{past}(\theta[\underline{\eta}])]$

c.



d. Disambiguation

- Reading 1: $\textsc{past}(\mathsf{want}'(j, {}^\wedge\textsc{past}(\mathsf{call}'(j))))$

$\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)]), \textsc{past}(\theta[\mathsf{call}'(j)]), \textsc{past}(\gamma[\mathsf{call}'(j)])]$
$\rightsquigarrow$(R3) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j), \textsc{past}(\theta[\mathsf{call}'(j)])]), \textsc{past}(\theta[\mathsf{call}'(j)]), \textsc{past}(\gamma[\mathsf{call}'(j)])]$
$\rightsquigarrow$(R2) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j), \textsc{past}(\theta[\mathsf{call}'(j)])]), \textsc{past}(\gamma[\mathsf{call}'(j)])]$
$\rightsquigarrow$(R3) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j), \textsc{past}(\theta[\mathsf{call}'(j)])]), \textsc{past}(\gamma[\mathsf{call}'(j), \mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j), \textsc{past}(\theta[\mathsf{call}'(j)])])])]$
$\rightsquigarrow$(R2) $\epsilon[\textsc{past}(\gamma[\mathsf{call}'(j), \mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j), \textsc{past}(\theta[\mathsf{call}'(j)])])])]$
$\rightsquigarrow$(R2, twice) $\epsilon[\textsc{past}(\gamma[\mathsf{want}'(j, {}^\wedge\zeta[\textsc{past}(\theta[\mathsf{call}'(j)])])])]$
$\rightsquigarrow$(R1, three times) $\textsc{past}(\mathsf{want}'(j, {}^\wedge\textsc{past}(\mathsf{call}'(j))))$

- Reading 2: $\mathsf{want}'(j, {}^\wedge\textsc{past}(\mathsf{call}'(j)))$

$\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)]), \textsc{past}(\theta[\mathsf{call}'(j)]), \textsc{past}(\gamma[\mathsf{call}'(j)])]$
$\rightsquigarrow$(R1, twice) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)]), \textsc{past}(\mathsf{call}'(j)), \textsc{past}(\mathsf{call}'(j))]$
$\rightsquigarrow$(R2) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)]), \textsc{past}(\mathsf{call}'(j))]$
$\rightsquigarrow$(R3) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j), \textsc{past}(\mathsf{call}'(j))]), \textsc{past}(\mathsf{call}'(j))]$
$\rightsquigarrow$(R2) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j), \textsc{past}(\mathsf{call}'(j))])]$
$\rightsquigarrow$(R2) $\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\textsc{past}(\mathsf{call}'(j))])]$
$\rightsquigarrow$(R1, twice) $\mathsf{want}'(j, {}^\wedge\textsc{past}(\mathsf{call}'(j)))$

- Reading 3: $\textsc{past}(\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j)))$

$\epsilon[\mathsf{want}'(j, {}^\wedge\zeta[\mathsf{call}'(j)]), \textsc{past}(\theta[\mathsf{call}'(j)]), \textsc{past}(\gamma[\mathsf{call}'(j)])]$
$\rightsquigarrow$(R1) $\epsilon[\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j)), \textsc{past}(\theta[\mathsf{call}'(j)]), \textsc{past}(\gamma[\mathsf{call}'(j)])]$
$\rightsquigarrow$(R3) $\epsilon[\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j)), \textsc{past}(\theta[\mathsf{call}'(j), \mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))]), \textsc{past}(\gamma[\mathsf{call}'(j)])]$
$\rightsquigarrow$(R3) $\epsilon[\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j)), \textsc{past}(\theta[\mathsf{call}'(j), \mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))]), \textsc{past}(\gamma[\mathsf{call}'(j), \mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))])]$
$\rightsquigarrow$(R2) $\epsilon[\textsc{past}(\theta[\mathsf{call}'(j), \mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))]), \textsc{past}(\gamma[\mathsf{call}'(j), \mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))])]$
$\rightsquigarrow$(R2, twice) $\epsilon[\textsc{past}(\theta[\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))]), \textsc{past}(\gamma[\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))])]$
$\rightsquigarrow$(R1, twice) $\epsilon[\textsc{past}(\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))), \textsc{past}(\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j)))]$
$\rightsquigarrow$(R2) $\epsilon[\textsc{past}(\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j)))]$
$\rightsquigarrow$(R1) $\textsc{past}(\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j)))$

4.1 Specific Constraints

Unavailable readings In addition to the three desired readings, we could also derive the following two readings:[1]

(32) a. $\mathsf{want}'(j, {}^\wedge\textsc{past}(\textsc{past}(\mathsf{call}'(j))))$

b. $\textsc{past}(\textsc{past}(\mathsf{want}'(j, {}^\wedge\mathsf{call}'(j))))$

In the present analysis, the two readings must be excluded with a principle.

(33) The Complex Tense Constraint (for Afrikaans):

For each verb with lrs $\langle\phi, \phi_-, \phi_\#\rangle$, there may be at most **one** temporal operator in $\phi_\#$ that has scope over the main semantic contribution of the verb.

For languages with a pluperfect, the Complex Tense Constraints would allow for two tense operators. In French and German, even a so called "double pluperfect" has been reported (Appuhn, 1966).

(34) (German) Maria hatte das Café schon verlassen gehabt.
Maria had.IMP the café already left.PART had.PART

Clause-boundedness of temporal operators

(35) Semantics Principle: Let $\langle\phi, \phi_-, \phi_\#\rangle$ be the lrs of the head daughter,
$\langle\psi, \psi_-, \psi_\#\rangle$ the lrs of the nonhead daughter,
and α a meta variable that does not occur in either lrs,

then the lrs of the mother results from eliminating every constraint from the following constraint lrs: $\langle\langle\alpha[\phi', \psi'], \phi_-, \phi_\#\rangle, \kappa\rangle$,

where κ contains exactly the following constraints:

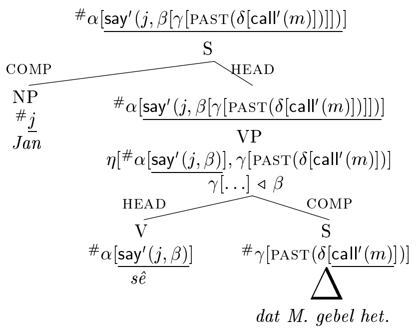
1. $\phi_\#$ is of the form $\beta[\vec{\phi}]$, and $\psi_\# \triangleleft \beta$ is in κ,
2. Specific Constraints:
   (a) If the non-head is a quantified NP complement, then its $\psi_\#$ is of the form $Qx[\psi' \circ \beta]$, for some quantifier Q, some logical connector $\circ$, some variable x, some expression ψ' and some meta variable β, and $\phi_- \triangleleft \gamma$ is in κ.
   (b) If the non-head is a raised complement of the head, then $\phi_- = \psi_-$ and $\phi_\# = \psi_\#$.
   (c) If the non-head is a complement clause, then ϕ_- is of the form $\mathsf{constant}'(x_1, \ldots, x_n, \alpha)$, and $\psi_\# \triangleleft \alpha$ is in κ.
   (d) ...

(36) a. Jan sê dat Marie gebel het.
Jan says.PRES that Marie called.PART AUX

b. *sê*: $^{\#}\alpha[\underline{\mathsf{say}'(j, \beta)}]$
het: $^{\#}\gamma[\textsc{past}(\delta[\underline{\epsilon}])]$
gebel: $^{\#}\zeta[\underline{\mathsf{call}'(m)}]$

[1] In a more elaborated version of the analysis (Sailer, 2003) in which we use explicit time variables in the logical forms, the readings in (32) are excluded due to extra conditions on the time variables.

c.



d. Disambiguation:
$\alpha[\mathsf{say}'(j, \beta[\gamma[\textsc{past}(\delta[\mathsf{call}'(m)])]])]$
$\leadsto$(R1, four times) $\mathsf{say}'(j, \textsc{past}(\mathsf{call}'(m)))$

Note: there is no way to derive a matrix-scope reading for the embedded tense.

5 Conclusion

- The system of past tense marking in Afrikaans provides empirical evidence for the need for scope underspecification and multiple exponents of semantic operators within the system of compositional semantics. Since LRS has incorporated these two features, it allows for a straightforward account of the data.
- Other instances of multiple exponents:
  - Interrogative operator (Richter and Sailer, 2001a):

    German: **Wer** hat **welches** Buch gelesen?
    who has which book read 'Who read which book?'
  - Negation operator (Richter and Sailer, 2001b, 2003):

    Polish: **Nikt** **nie** pomaga **nikomu**.
    nobody NEG helped nobody 'Nobody helped anybody.'
- All parts of LRS are independently necessary.
- We provided an HPSG-neutral characterization of LRS.

Future work

- on Afrikaans:
  - more detailed empirical study
  - including: sequence-of-tense, extraposed infinitival clauses, adjunct clauses
  - analysis of temporal adverbs
- on LRS:
  - study of other instances of multiple exponents of semantic operators
  - detailed comparison between LRS and CLLS (Egg et al., 2001)
  - implementation of an LRS module (joint work with Gerald Penn and Frank Richter)

References

Appuhn, Hans-Günther (1966). Die „hyperperiphrastischen Tempora" im Französischen und Deutschen. *Die neueren Sprachen 15*, 237–243.

Bos, Johan (1996). Predicate Logic Unplugged. In P. Dekker and M. Stokhof (Eds.), *Proceedings of the Tenth Amsterdam Colloquium*, pp. 133–143. ILLC/Department of Philosophy, University of Amsterdam.

Bouma, Gosse (2003). Verb Clusters and the Scope of Adjuncts in Dutch. In P. A. M. Seuren and G. Kempen (Eds.), *Verb Constructions in German and Dutch*. Amsterdam and Philadelphia: Benjamins.

Villiers, Meyer de (1971). *Die grammatika van tyd en modaliteit* (2nd, revised ed.). Kaapstad: Balkema.

Donaldson, Bruce C. (1993). *A Grammar of Afrikaans*. Berlin/New York: Mouton de Gruyter.

Egg, Markus, Koller, Alexander, and Niehren, Joachim (2001). The Constraint Language for Lambda Structures. *Journal of Logic, Language and Information 10*(4), 457–485.

Kleij, Susanne van der (1999). Tijd en aspect in het werkwoordelijke systeem van het Afrikaans. M.A. thesis, University of Leiden.

Pollard, Carl and Sag, Ivan A. (1994). *Head-Driven Phrase Structure Grammar*. University of Chicago Press, Chicago and London.

Ponelis, F. A. (1979). *Afrikaanse Sintaksis*. van Schaik, Johannesburg.

Reyle, Uwe (1993). Dealing with Ambiguities by Underspecification: Construction, Representation and Deduction. *Journal of Semantics 10*(2), 123–179.

Richter, Frank and Sailer, Manfred (2001a). On the Left Periphery of German finite Sentences. In W. D. Meurers and T. Kiss (Eds.), *Constraint-Based Approaches to Germanic Syntax*, pp. 257–300. Stanford: CSLI Publications.

Richter, Frank and Sailer, Manfred (2001b). Polish Negation and Lexical Resource Semantics. In G.-J. Kruijff, L. S. Moss, and R. T. Oehrle (Eds.), *Proceedings FG-MOL 2001*, Number 53 in Electronic Notes in Theoretical Computer Science. Elsevier Science.

Richter, Frank and Sailer, Manfred (2003). Basic Concepts of Lexical Resource Semantics. Lecture notes for the course on *Constraint-based Combinatorial Semantics*, ESSLLI 2003, Vienna.

Sailer, Manfred (2003). Past Tense Marking in Afrikaans. Handout to a presentation given at the *HPSG-Workshop on Germanic*, University of Bremen, August 7–8, 2003.