Overriding default orderings in inflectional morphology

Gregory Stump
University of Kentucky
gstump@uky.edu

[*Conference on Defaults in Morphological Theory, University of Kentucky, May 21-22, 2012*]

1. Two types of default ordering in inflectional morphology

- the linear order of an affix with respect to the stem to which it attaches
- the linear order in which blocks of realization rules apply

1. Two types of default ordering in inflectional morphology

- the linear order of an affix with respect to the stem to which it attaches
- the linear order in which blocks of realization rules apply

1. Two types of default ordering in inflectional morphology

- the linear order of an affix with respect to the stem to which it attaches
- the linear order in which blocks of realization rules apply

Both sorts of order may be regulated by defaults and overrides:

- ***default-affixation overrides*** are conditioned by ***affixation-inversion property sets***
- ***default-composition overrides*** are conditioned by ***composition-inversion property sets***

Both sorts of order may be regulated by defaults and overrides:

- ***default-affixation overrides*** are conditioned by ***affixation-inversion property sets***
- ***default-composition overrides*** are conditioned by ***composition-inversion property sets***

Both sorts of order may be regulated by defaults and overrides:

- ***default-affixation overrides*** are conditioned by ***affixation-inversion property sets***
- ***default-composition overrides*** are conditioned by ***composition-inversion property sets***

Both sorts of order may be regulated by defaults and overrides:

- ***default-affixation overrides*** are conditioned by ***affixation-inversion property sets***
- ***default-composition overrides*** are conditioned by ***composition-inversion property sets***

Both sorts of order may be regulated by defaults and overrides:

- ***default-affixation overrides*** are conditioned by ***affixation-inversion property sets***
- ***default-composition overrides*** are conditioned by ***composition-inversion property sets***

Talk outline

1. Examples of default ordering
2. Three default-ordering operators
3. French pronominal clitics
4. A PFM analysis of French pronominal clitics
5. Two concluding remarks

Talk outline

1. Examples of default ordering
2. Three default-ordering operators
3. French pronominal clitics
4. A PFM analysis of French pronominal clitics
5. Two concluding remarks

Talk outline

1. Examples of default ordering
2. Three default-ordering operators
3. French pronominal clitics
4. A PFM analysis of French pronominal clitics
5. Two concluding remarks

Talk outline

1. Examples of default ordering
2. Three default-ordering operators
3. French pronominal clitics
4. A PFM analysis of French pronominal clitics
5. Two concluding remarks

Talk outline

1. Examples of default ordering
2. Three default-ordering operators
3. French pronominal clitics
4. A PFM analysis of French pronominal clitics
5. Two concluding remarks

1.1 A default-affixation override: Swahili relative affixes

(1) Swahili relative affixes

GENDER	SINGULAR	PLURAL
m/wa	*ye*	*o*
m/mi	*o*	*yo*
ki/vi	*cho*	*vyo*
ji/ma	*lo*	*yo*
n/n	*yo*	*zo*
u	*o*	—
u/n	*o*	*zo*
ku	*ko*	—

Swahili relative affixes are prefixed by default, but are suffixed in tenseless affirmative forms

(2a)

a–na–vyo–vi–soma
SUBJ:CL.m–TNS–REL:CL.vi–OBJ:CL.vi–read
‘(books) which he is reading’

(2a)

a–na–vyo–vi–soma

SUBJ:CL.m–TNS–REL:CL.vi–OBJ:CL.vi–read

‘(books) which he is reading’

(2a)

a–na–vyo–vi–soma

SUBJ:CL.m–TNS–REL:CL.vi–OBJ:CL.vi–read

‘(books) which he is reading’

(2b)

a–si–vyo–vi–taka
SUBJ:CL.m–NEG–REL:CL.vi–OBJ:CL.vi–want
‘(books) which he doesn’t want’

(2b)

a–si–vyo–vi–taka

SUBJ:CL.m–NEG–REL:CL.vi–OBJ:CL.vi–want

‘(books) which he doesn’t want’

(2b)

a–si–vyo–vi–taka

SUBJ:CL.m–NEG–REL:CL.vi–OBJ:CL.vi–want

‘(books) which he doesn’t want’

(2c)

a–vi–taka–vyo
SUBJ:CL.m–OBJ:CL.vi–want–REL:CL.vi
‘(books) which he wants’

(2c)

a–vi–taka–vyo

SUBJ:CL.m–OBJ:CL.vi–want–REL:CL.vi

‘(books) which he wants’

(2c)

a–vi–taka–vyo

SUBJ:CL.m–OBJ:CL.vi–want–REL:CL.vi

‘(books) which he wants’

1.2 A default-composition override: Fula subject and object agreement

In Fula relative past tense verb forms, subject agreement precedes object agreement by default;

this default is overridden in forms in which 1sg subject agreement coincides with 2sg or 3sg.class.1 object agreement

In Fula relative past tense verb forms, subject agreement precedes object agreement by default;

this default is overridden in forms in which 1sg subject agreement coincides with 2sg or 3sg.class.1 object agreement

(3a)

mball–u–mi–ɓe’
help–REL.PAST.ACT–SUBJ:I–OBJ:them.CL.2
‘I helped them’

(3a)

*mball–u–**mi**–be’*

help–REL.PAST.ACT–SUBJ:I–OBJ:them.CL.2

‘I helped them’

(3a)

mball–u–mi–ɓe’

help–REL.PAST.ACT–SUBJ:I–OBJ:them.CL.2

‘I helped them’

(3b)

mball–u–ɗaa–mo’
help–REL.PAST.ACT–SUBJ:you–OBJ:him.CL.1
‘you (sg.) helped him’

(3b)

mball–u–ɗaa–mo’

help–REL.PAST.ACT–SUBJ:**you**–OBJ:him.CL.1

‘you (sg.) helped him’

(3b)

mball–u–ɗaa–mo’

help–REL.PAST.ACT–SUBJ:you–OBJ:him.CL.1

‘you (sg.) helped him’

(3c)

mball–u–maa–mi’
help–REL.PAST.ACT–OBJ:you–SUBJ:I
‘I helped you (sg.)’

(3c)

mball–u–maa–mi’
help–REL.PAST.ACT–OBJ:you–SUBJ:I
‘I helped you (sg.)’

(3c)

mball–u–maa–mi’
help–REL.PAST.ACT–OBJ:you–SUBJ:I
‘I helped you (sg.)’

(3d)

mball–u–moo–mi’
help–REL.PAST.ACT–OBJ:him.CL.1–SUBJ:I
‘I helped him’

(3d)

mball–u–moo–mi’
help–REL.PAST.ACT–OBJ:him.CL.1–SUBJ:I
‘I helped him’

(3d)

mball–u–moo–mi’
help–REL.PAST.ACT–OBJ:him.CL.1–SUBJ:I
‘I helped him’

2. Three default-ordering operators

(4a) Default-prefixation operator [<]

$$[<\mathbf{S}, X]$$

set of affixation-inversion property sets

affix

(4a) Default-prefixation operator [<]

$$[<S, X]$$

set of affixation-inversion property sets

affix

(4a) Default-prefixation operator [<]

$$[<S, X]$$

set of affixation-inversion property sets

affix

(4a) Default-prefixation operator [<]

[< S, X]

applies to realize the pairing ⟨Y, σ⟩
of a stem Y with a morphosyntactic
property set σ

(4a) Default-prefixation operator [<]

$$[<\mathbf{S}, X](\langle Y, \sigma\rangle) = \langle YX, \sigma\rangle \text{ if } \sigma \in \mathbf{S};$$
$$= \langle XY, \sigma\rangle \text{ otherwise.}$$

(4a) Default-prefixation operator [<]

$[<\mathbf{S}, X](\langle Y, \sigma\rangle) = \langle YX, \sigma\rangle$ if $\sigma \in \mathbf{S}$;
$= \langle XY, \sigma\rangle$ otherwise.

(4a) Default-prefixation operator [<]

$$[<\mathbf{S}, X](\langle Y, \sigma\rangle) = \langle YX, \sigma\rangle \text{ if } \sigma \in \mathbf{S};$$
$$= \langle XY, \sigma\rangle \text{ otherwise.}$$

(4b) Default-suffixation operator [>]

(analogous to [<])

[<] and Swahili relative affixes

(5) ***Paradigm function*** for
Swahili relative verb forms

Where σ is a morphosyntactic property set and X is the root of verbal lexeme L,

$$\text{PF}(\langle \text{L}, \sigma \rangle) = [\text{IV} : [\text{III} : [\text{II} : [\text{I} : \langle \text{X}, \sigma \rangle]]]]$$

(6) ***Nar notation***

[*n* : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in Block *n* to ⟨Y, σ⟩

(5) ***Paradigm function*** for
Swahili relative verb forms

Where σ is a morphosyntactic property set and X is the root of verbal lexeme L,

$$\mathrm{PF}(\langle \mathrm{L}, \sigma \rangle) = [\mathrm{IV} : [\mathrm{III} : [\mathrm{II} : [\mathrm{I} : \langle \mathrm{X}, \sigma \rangle]]]]$$

(6) ***Nar notation***

[*n* : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in Block *n* to ⟨Y, σ⟩

(5) ***Paradigm function*** for
Swahili relative verb forms

Where σ is a morphosyntactic property set and X is the root of verbal lexeme L,

$$PF(\langle L, \sigma \rangle) = [IV : [III : [II : [I : \langle X, \sigma \rangle]]]]$$

(6) ***Nar notation***

[*n* : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in Block *n* to ⟨Y, σ⟩

(5) ***Paradigm function*** for
Swahili relative verb forms

Where σ is a morphosyntactic property set and X is the root of verbal lexeme L,

$$PF(\langle L, \sigma \rangle) = [IV : [III : [II : [I : \langle X, \sigma \rangle]]]]$$

(6) ***Nar notation***

[*n* : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in Block *n* to ⟨Y, σ⟩

(5) ***Paradigm function*** for
Swahili relative verb forms

Where σ is a morphosyntactic property set and X is the root of verbal lexeme L,

$$PF(\langle L, \sigma \rangle) = [IV : [III : [II : [I : \langle X, \sigma \rangle]]]]$$

(6) ***Nar notation***

[*n* : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in Block *n* to ⟨Y, σ⟩

(5) ***Paradigm function*** for
Swahili relative verb forms

Where σ is a morphosyntactic property set and X is the root of verbal lexeme L,

$$PF(\langle L, \sigma \rangle) = [IV : [III : [II : [I : \langle X, \sigma \rangle]]]]$$

(6) ***Nar notation***

[*n* : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in Block *n* to ⟨Y, σ⟩

(7) ***Realization rules***
for Swahili relative verb forms

a. **Block I** (object agreement)
b. **Block II** (relative affixes)
c. **Block III** (tense, polarity)
d. **Block IV** (subject agreement)

(7) ***Realization rules***
for Swahili relative verb forms

a. **Block I** (object agreement)

b. **Block II** (relative affixes)

c. **Block III** (tense, polarity)

d. **Block IV** (subject agreement)

(7) ***Realization rules***
for Swahili relative verb forms

a. **Block I** (object agreement)

b. **Block II** (relative affixes)

c. **Block III** (tense, polarity)

d. **Block IV** (subject agreement)

(7) ***Realization rules***
for Swahili relative verb forms

a. **Block I** (object agreement)

b. **Block II** (relative affixes)

c. **Block III** (tense, polarity)

d. **Block IV** (subject agreement)

(7) ***Realization rules***
for Swahili relative verb forms

a. **Block I** (object agreement)

b. **Block II** (relative affixes)

c. **Block III** (tense, polarity)

d. **Block IV** (subject agreement)

(7b) **Block II** (in which μ = su or ob)

V,{REL(μ):{sg, m/wa}}: ⟨X, σ⟩ → [<**S**, *ye*](⟨X, σ⟩)
V,{REL(μ):{sg, ki/vi}}: ⟨X, σ⟩ → [<**S**, *cho*](⟨X, σ⟩)
V,{REL(μ):{sg, n/n}}: ⟨X, σ⟩ → [<**S**, *yo*](⟨X, σ⟩)
V,{REL(μ):{pl, m/wa}}: ⟨X, σ⟩ → [<**S**, *o*](⟨X, σ⟩)
V,{REL(μ):{pl, ki/vi}}: ⟨X, σ⟩ → [<**S**, *vyo*](⟨X, σ⟩)
V,{REL(μ):{pl, n/n}}: ⟨X, σ⟩ → [<**S**, *zo*](⟨X, σ⟩)

(7b) **Block II** (in which μ = su or ob)

V,{REL(μ):{sg, m/wa}}: ⟨X, σ⟩ → [<**S**, *ye*](⟨X, σ⟩)

V,{REL(μ):{sg, ki/vi}}: ⟨X, σ⟩ → [<**S**, *cho*](⟨X, σ⟩)

V,{REL(μ):{sg, n/n}}: ⟨X, σ⟩ → [<**S**, *yo*](⟨X, σ⟩)

V,{REL(μ):{pl, m/wa}}: ⟨X, σ⟩ → [<**S**, *o*](⟨X, σ⟩)

V,{REL(μ):{pl, ki/vi}}: ⟨X, σ⟩ → [<**S**, *vyo*](⟨X, σ⟩)

V,{REL(μ):{pl, n/n}}: ⟨X, σ⟩ → [<**S**, *zo*](⟨X, σ⟩)

(8) ***Affixation-inversion property sets*** in Swahili

S is the set of only and all tenseless affirmative property sets

(8) ***Affixation-inversion property sets*** in Swahili

S is the set of only and all tenseless affirmative property sets

By default:

$$[<\boldsymbol{S}, \mathit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \mathit{vyo}\mathrm{Y}, \sigma \rangle$$

Where σ is an affixation-inversion property set:

$$[<\boldsymbol{S}, \mathit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \mathrm{Y}\mathit{vyo}, \sigma \rangle$$

By default:

$$[<\mathbf{S}, \mathit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \mathit{vyo}\mathrm{Y}, \sigma \rangle$$

Where σ is an affixation-inversion property set:

$$[<\mathbf{S}, \mathit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \mathrm{Y}\mathit{vyo}, \sigma \rangle$$

By default:

$$[<\mathbf{S}, \mathit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \mathit{vyo}\mathrm{Y}, \sigma \rangle$$

Where σ is an affixation-inversion property set:

$$[<\mathbf{S}, \mathit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \mathrm{Y}\mathit{vyo}, \sigma \rangle$$

By default:

$$[<\mathbf{S}, \textit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \textit{vyo}\mathrm{Y}, \sigma \rangle$$

Where σ is an affixation-inversion property set:

$$[<\mathbf{S}, \textit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \mathrm{Y}\textit{vyo}, \sigma \rangle$$

By default:

$$[<\mathbf{S}, \textit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \textit{vyo}\mathrm{Y}, \sigma \rangle$$

Where σ is an affixation-inversion property set:

$$[<\mathbf{S}, \textit{vyo}](\langle \mathrm{Y}, \sigma \rangle) = \langle \mathrm{Y}\textit{vyo}, \sigma \rangle$$

(9) Proof of *a–na–vyo–vi–soma*

‘(books) which he is reading’

(10) Proof of *a–vi–taka–vyo*

‘(books) which he wants’

(11) Default-composition operator [○]

a. Recursive definition of ***block construct***
   i. If m is a rule block, then m is a block construct.
   ii. If m, n are block constructs and $\mathbf{S}$ is a set of composition-inversion property sets, then $[\circ \mathbf{S}, m, n]$ is a block construct.

(11) Default-composition operator [○]

a. Recursive definition of ***block construct***

   i. If m is a rule block, then m is a block construct.

   ii. If m, n are block constructs and $\mathbf{S}$ is a set of composition-inversion property sets, then $[\circ \mathbf{S}, m, n]$ is a block construct.

(11) Default-composition operator [○]

a. Recursive definition of ***block construct***
   i. If m is a rule block, then m is a block construct.
   ii. If m, n are block constructs and $\mathbf{S}$ is a set of composition-inversion property sets, then $[\circ \mathbf{S}, m, n]$ is a block construct.

(11) Default-composition operator [○]

a. Recursive definition of ***block construct***
   i. If *m* is a rule block, then *m* is a block construct.
   ii. If *m*, *n* are block constructs and **S** is a set of composition-inversion property sets, then [○**S**, *m*, *n*] is a block construct.

(11) Default-composition operator [○]

b. ***Nar*** [narrowest applicable rule] ***notation*** (revised)

Given block construct BC and pairing ⟨Y, σ⟩:

- if BC is a block construct [○**S**, *m*, *n*],
  [BC : ⟨Y, σ⟩] = [*n* : [*m* : ⟨Y, σ⟩]] if σ ∈ **S**,
  = [*m* : [*n* : ⟨Y, σ⟩]] otherwise;
- if BC is a rule block, [BC : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in BC to ⟨Y, σ⟩.

(11) Default-composition operator [○]

b. ***Nar*** [narrowest applicable rule] ***notation*** (revised)

Given block construct BC and pairing ⟨Y, σ⟩:

- if BC is a block construct [○**S**, *m*, *n*],
  [BC : ⟨Y, σ⟩] = [*n* : [*m* : ⟨Y, σ⟩]] if σ ∈ **S**,
  = [*m* : [*n* : ⟨Y, σ⟩]] otherwise;
- if BC is a rule block, [BC : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in BC to ⟨Y, σ⟩.

(11) Default-composition operator [○]

b. ***Nar*** [narrowest applicable rule] ***notation*** (revised)

Given block construct BC and pairing ⟨Y, σ⟩:

- if BC is a block construct [○**S**, *m*, *n*],
  [BC : ⟨Y, σ⟩] = [*n* : [*m* : ⟨Y, σ⟩]] if σ ∈ **S**,
  = [*m* : [*n* : ⟨Y, σ⟩]] otherwise;
- if BC is a rule block, [BC : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in BC to ⟨Y, σ⟩.

(11) Default-composition operator [○]

b. ***Nar*** [narrowest applicable rule] ***notation*** (revised)

Given block construct BC and pairing ⟨Y, σ⟩:

- if BC is a block construct [○**S**, *m*, *n*],
  [BC : ⟨Y, σ⟩] = [*n* : [*m* : ⟨Y, σ⟩]] if σ ∈ **S**,
  = [*m* : [*n* : ⟨Y, σ⟩]] otherwise;
- if BC is a rule block, [BC : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in BC to ⟨Y, σ⟩.

(11) Default-composition operator [○]

b. ***Nar*** [narrowest applicable rule] ***notation*** (revised)

Given block construct BC and pairing ⟨Y, σ⟩:

- if BC is a block construct [○**S**, *m*, *n*],
  [BC : ⟨Y, σ⟩] = [*n* : [*m* : ⟨Y, σ⟩]] if σ ∈ **S**,
  = [*m* : [*n* : ⟨Y, σ⟩]] otherwise;
- if BC is a rule block, [BC : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in BC to ⟨Y, σ⟩.

(11) Default-composition operator [○]

b. ***Nar*** [narrowest applicable rule] ***notation*** (revised)

Given block construct BC and pairing ⟨Y, σ⟩:

- if BC is a block construct [○**S**, *m*, *n*],
  [BC : ⟨Y, σ⟩] = [*n* : [*m* : ⟨Y, σ⟩]] if σ ∈ **S**,
  = [*m* : [*n* : ⟨Y, σ⟩]] otherwise;
- if BC is a rule block, [BC : ⟨Y, σ⟩] is the result of applying the narrowest applicable rule in BC to ⟨Y, σ⟩.

[○] and Fula verb agreement

(12) ***Realization rules*** for relative past tense verb forms in Fula

a. **Block TENSE**

b. **Block SUBJ**

c. **Block OBJ**

(12) ***Realization rules*** for relative past tense verb forms in Fula

a. **Block TENSE**

b. **Block SUBJ**

c. **Block OBJ**

(12) ***Realization rules*** for relative past tense verb forms in Fula

a. **Block TENSE**

b. **Block SUBJ**

c. **Block OBJ**

(12) ***Realization rules*** for relative past tense verb forms in Fula

a. **Block TENSE**

b. **Block SUBJ**

c. **Block OBJ**

(13)

Paradigm function for relative past tense verb forms in Fula

PF(⟨L, σ⟩)

$$= [[\circ\mathbf{S}, \text{OBJ}, \text{SUBJ}] : [\text{TENSE} : \langle X, \sigma \rangle]]$$

(14) *Composition-inversion property sets* in Fula

S =

{ {AGR(su):{1sg}, rel. past act., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past act., AGR(ob): {3sg, CL.1}},
{AGR(su):{1sg}, rel. past mid., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past mid., AGR(ob): {3sg, CL.1}} }

(14) ***Composition-inversion property sets*** in Fula

S =

{ {AGR(su):{1sg}, rel. past act., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past act., AGR(ob): {3sg, CL.1}},
{AGR(su):{1sg}, rel. past mid., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past mid., AGR(ob): {3sg, CL.1}} }

(14) *Composition-inversion property sets* in Fula

S =

{ {AGR(su):{1sg}, rel. past act., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past act., AGR(ob): {3sg, CL.1}},
{AGR(su):{1sg}, rel. past mid., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past mid., AGR(ob): {3sg, CL.1}} }

(14) *Composition-inversion property sets* in Fula

S =

{ {AGR(su):{1sg}, rel. past act., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past act., AGR(ob): {3sg, CL.1}},
{AGR(su):{1sg}, rel. past mid., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past mid., AGR(ob): {3sg, CL.1}} }

(14) *Composition-inversion property sets* in Fula

S =

{ {AGR(su):{1sg}, rel. past act., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past act., AGR(ob): {3sg, CL.1}},
{AGR(su):{1sg}, rel. past mid., AGR(ob):{2sg}},
{AGR(su):{1sg}, rel. past mid., AGR(ob): {3sg, CL.1}} }

By default:

$$[[\circ\mathbf{S}, \textsc{obj}, \textsc{subj}] : \langle X, \sigma \rangle]$$
$$= [\textsc{obj} : [\textsc{subj} : \langle X, \sigma \rangle]]$$

Where σ is an affixation-inversion property set:

$$[[\circ\mathbf{S}, \textsc{obj}, \textsc{subj}] : \langle X, \sigma \rangle]$$
$$= [\textsc{subj} : [\textsc{obj} : \langle X, \sigma \rangle]]$$

By default:

$$[[\circ\mathbf{S}, \text{OBJ}, \text{SUBJ}] : \langle X, \sigma \rangle] = [\text{OBJ} : [\text{SUBJ} : \langle X, \sigma \rangle]]$$

Where σ is an affixation-inversion property set:

$$[[\circ\mathbf{S}, \text{OBJ}, \text{SUBJ}] : \langle X, \sigma \rangle] = [\text{SUBJ} : [\text{OBJ} : \langle X, \sigma \rangle]]$$

By default:

$$[[\circ\mathbf{S}, \text{OBJ}, \text{SUBJ}] : \langle X, \sigma \rangle]$$
$$= [\text{OBJ} : [\text{SUBJ} : \langle X, \sigma \rangle]]$$

Where σ is an affixation-inversion property set:

$$[[\circ\mathbf{S}, \text{OBJ}, \text{SUBJ}] : \langle X, \sigma \rangle]$$
$$= [\text{SUBJ} : [\text{OBJ} : \langle X, \sigma \rangle]]$$

By default:

$$[[\circ\mathbf{S}, \text{OBJ}, \text{SUBJ}] : \langle X, \sigma \rangle] = [\text{OBJ} : [\text{SUBJ} : \langle X, \sigma \rangle]]$$

Where σ is an affixation-inversion property set:

$$[[\circ\mathbf{S}, \text{OBJ}, \text{SUBJ}] : \langle X, \sigma \rangle] = [\text{SUBJ} : [\text{OBJ} : \langle X, \sigma \rangle]]$$

(15) Proof of *mball–u–mi–6e'*
'I helped them'

(16) Proof of *mball–u–moo–mi'*
'I helped him'

3. French pronominal clitics

(17) Basic template for proclitic local-argument pronouns

A (ACC or DAT)	B (ACC)	C (DAT)
me	*le*	*lui*
te	*la*	*leur*
se	*les*	
nous		
vous		

(17) Basic template for proclitic local-argument pronouns

A (ACC or DAT)	B (ACC)	C (DAT)
me	*le*	*lui*
te	*la*	*leur*
se	*les*	
nous		
vous		

(18) *Jean* { *me* / *te* / *se* / *le* / *la* / *nous* / *vous* / *les* } *verra.*

(19) *Jean* { *me* / *te* / *se* / *lui* / *nous* / *vous* / *leur* } *donnera le livre.*

(20) *Jean* { *me* / *te* / *se* / *nous* / *vous* } { *le* / *la* / *les* } *présentera.*

(21) *Jean* { *le* / *la* / *les* } { *lui* / *leur* } *présentera.*

(22) a. *Jean me te présentera.

b. Jean me présentera à toi.

c. Jean te présentera à moi.

(22) a. **Jean me te présentera.*
b. *Jean me présentera à toi.*
c. *Jean te présentera à moi.*

(23) *Jean { *me* / *te* / *se* / *nous* / *vous* } { *lui* / *leur* } *présentera.*

(24) *Jean me présentera à lui.*

(23) *Jean { *me* / *te* / *se* / *nous* / *vous* } { *lui* / *leur* } *présentera.*

(24) *Jean me présentera à lui.*

(25) Basic template for enclitic local-argument pronouns

D (ACC)	E (ACC or DAT; 3rd person DAT only)
le	*moi*
la	*toi*
les	*lui*
	nous
	vous
	leur

(26) a. *Présentez* - { *le* / *la* / *les* } - { *moi!* / *lui!* / *nous!* / *leur!* }

b. *Dépêche-toi!*

c. *Dépêchez-vous!*

(27) Clitic climbing

a. *Marie le lui a fait manger.*

‘Marie made him eat it.’

b. *Jean me la fait envoyer.*

‘Jean has me send it.’

or ‘Jean has it sent to me.’

c. *Pierre la lui a fait écrire par Jean.*

‘Pierre made Jean write it to him.’

Extra datives: ethical datives

(28) *Il te nous a passé un de ces savons!*
'He gave us an incredible telling-off!'

(29) *%Nous ne nous vous le faisons pas dire.*
'We couldn't agree more, you don't need to tell us.'
[Literally: 'We don't make you say it, we really don't.']

Extra datives: ethical datives

(28) *Il te nous a passé un de ces savons!*
'He gave us an incredible telling-off!'

(29) *%Nous ne nous vous le faisons pas dire.*
'We couldn't agree more, you don't need to tell us.'
[Literally: 'We don't make you say it, we really don't.']

Extra datives: ethical datives

(28) *Il te nous a passé un de ces savons!*
'He gave us an incredible telling-off!'

(29) %*Nous ne nous vous le faisons pas dire.*
'We couldn't agree more, you don't need to tell us.'
[Literally: 'We don't make you say it, we really don't.']

(30) Extra datives: clitic climbing

a. *Elle me te les a fait envoyer.*
'She made me send them to you.'

b. *Elle me les lui a fait envoyer.*
'She made me send them to her.'

c. **Elle leur les lui a fait envoyer.*
Putatively:
'She made them send them to her.'

(30) Extra datives: clitic climbing

a. *Elle me te les a fait envoyer.*
'She made me send them to you.'

b. *Elle me les lui a fait envoyer.*
'She made me send them to her.'

c. **Elle leur les lui a fait envoyer.*
Putatively:
'She made them send them to her.'

(30) Extra datives: clitic climbing

a. *Elle me te les a fait envoyer.*
'She made me send them to you.'

b. *Elle me les lui a fait envoyer.*
'She made me send them to her.'

c. ***Elle leur les lui a fait envoyer.***
Putatively:
'She made them send them to her.'

(31) Extra datives: clitic climbing

a. %*Jean me te semble fidèle.*
'Jean seems to me to be faithful to you.'

b. %*Jean me lui semble fidèle.*
'Jean seems to me to be faithful to her.'

c. **Jean lui leur semble fidèle.*
Putatively:
'Jean seems to her to be faithful to them.'

(31) Extra datives: clitic climbing

a. %*Jean me te semble fidèle.*
'Jean seems to me to be faithful to you.'

b. %*Jean me lui semble fidèle.*
'Jean seems to me to be faithful to her.'

c. **Jean lui leur semble fidèle.*
Putatively:
'Jean seems to her to be faithful to them.'

(32) Expanded proclitic template for non-local-argument as well as local-argument pronouns

A′ (“extra” DAT)	A (ACC or DAT)	B (ACC)	C (DAT)
me	*me*	*le*	*lui*
te	*te*	*la*	*leur*
se	*se*	*les*	
nous	*nous*		
vous	*vous*		

4. A PFM analysis of French pronominal clitics: Defaults and overrides

4.1 Preliminary assumptions

(35) Synthetic property sets:
Sets of specifications
of AGR and TAM

(36) Clitic property sets:
Sets of specifications of POL,
pronominal ACC
and pronominal DAT

(35) Synthetic property sets:
Sets of specifications
of AGR and TAM

(36) Clitic property sets:
Sets of specifications of POL,
pronominal ACC
and pronominal DAT

(37)

	Synthetic paradigm of DONNER	Clitic-group paradigm of DONNER
Sample cell	⟨DONNER, {AGR:{3sg}, fut}⟩	⟨*donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩
Its realization	⟨*donnera*, {AGR:{3sg}, fut}⟩	⟨*le lui donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩

(37)

	Synthetic paradigm of DONNER	Clitic-group paradigm of DONNER
Sample cell	⟨DONNER, {AGR:{3sg}, fut}⟩	⟨*donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩ ⟨*le lui donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩
Its realization	⟨*donnera*, {AGR:{3sg}, fut}⟩	

(37)

	Synthetic paradigm of DONNER	Clitic-group paradigm of DONNER
Sample cell	⟨DONNER, {AGR:{3sg}, fut}⟩	⟨*donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩
Its realization	⟨*donnera*, {AGR:{3sg}, fut}⟩	⟨*le lui donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩

(37)

	Synthetic paradigm of DONNER	Clitic-group paradigm of DONNER
Sample cell	⟨DONNER, {AGR:{3sg}, fut}⟩	⟨*donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩
Its realization	⟨*donnera*, {AGR:{3sg}, fut}⟩	⟨*le lui donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩

(37)

	Synthetic paradigm of DONNER	Clitic-group paradigm of DONNER
Sample cell	⟨DONNER, {AGR:{3sg}, fut}⟩	⟨*donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩
Its realization	⟨*donnera*, {AGR:{3sg}, fut}⟩	⟨*le lui donnera*, {AGR:{3sg}, fut} ∪ {affirm, ACC:{3sgm}, DAT:{3sg}}⟩

4.2 Realizing clitic-group paradigms

(38) ***Realization rules*** for French pronominal clitics

a. **Block I.**

V,{ACC:{3sg, masc, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{le}](\langle X, \sigma \rangle)$
V,{ACC:{3sg, fem, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{la}](\langle X, \sigma \rangle)$
V,{ACC:{3pl, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{les}](\langle X, \sigma \rangle)$

b. **Block II.** Where μ = ACC or DAT,

V,{μ:{1sg}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{me}](\langle X, \sigma \rangle)$
V,{μ:{2sg}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{te}](\langle X, \sigma \rangle)$
V,{μ:{3, refl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{se}](\langle X, \sigma \rangle)$
V,{μ:{1pl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{nous}](\langle X, \sigma \rangle)$
V,{μ:{2pl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{vous}](\langle X, \sigma \rangle)$
V,{DAT:{3sg, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{lui}](\langle X, \sigma \rangle)$
V,{DAT:{3pl, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{leur}](\langle X, \sigma \rangle)$

(38) ***Realization rules*** for French pronominal clitics

a. **Block I.**

V,{ACC:{3sg, masc, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{le}](\langle X, \sigma \rangle)$
V,{ACC:{3sg, fem, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{la}](\langle X, \sigma \rangle)$
V,{ACC:{3pl, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{les}](\langle X, \sigma \rangle)$

b. **Block II.** Where μ = ACC or DAT,

V,{μ:{1sg}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{me}](\langle X, \sigma \rangle)$
V,{μ:{2sg}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{te}](\langle X, \sigma \rangle)$
V,{μ:{3, refl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{se}](\langle X, \sigma \rangle)$
V,{μ:{1pl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{nous}](\langle X, \sigma \rangle)$
V,{μ:{2pl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{vous}](\langle X, \sigma \rangle)$
V,{DAT:{3sg, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{lui}](\langle X, \sigma \rangle)$
V,{DAT:{3pl, nonrefl}}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{leur}](\langle X, \sigma \rangle)$

(38) ***Realization rules*** for French pronominal clitics

c. **Block III.**

V, {DAT2:τ}: ⟨X, σ⟩ → ⟨Y, σ⟩,
where [II : ⟨X, {DAT:τ}⟩] = ⟨Y, {DAT:τ}⟩

b. **Block IV.**

V, {neg}: ⟨X, σ⟩ → ⟨*ne* X, σ⟩

(39) The set $\mathbf{S}_1$ of ***affixation-inversion property sets*** for (38a,b):

$\mathbf{S}_1$ is the set of affirmative imperative property sets

(39) The set $\mathbf{S}_1$ of ***affixation-inversion property sets*** for (38a,b):

$\mathbf{S}_1$ is the set of affirmative imperative property sets

(40) Clause in the definition
of the French ***paradigm function***

PF(⟨W, σ⟩)
= [IV : [III : [[∘**S**$_2$, II, I] : ⟨W, σ⟩]]]

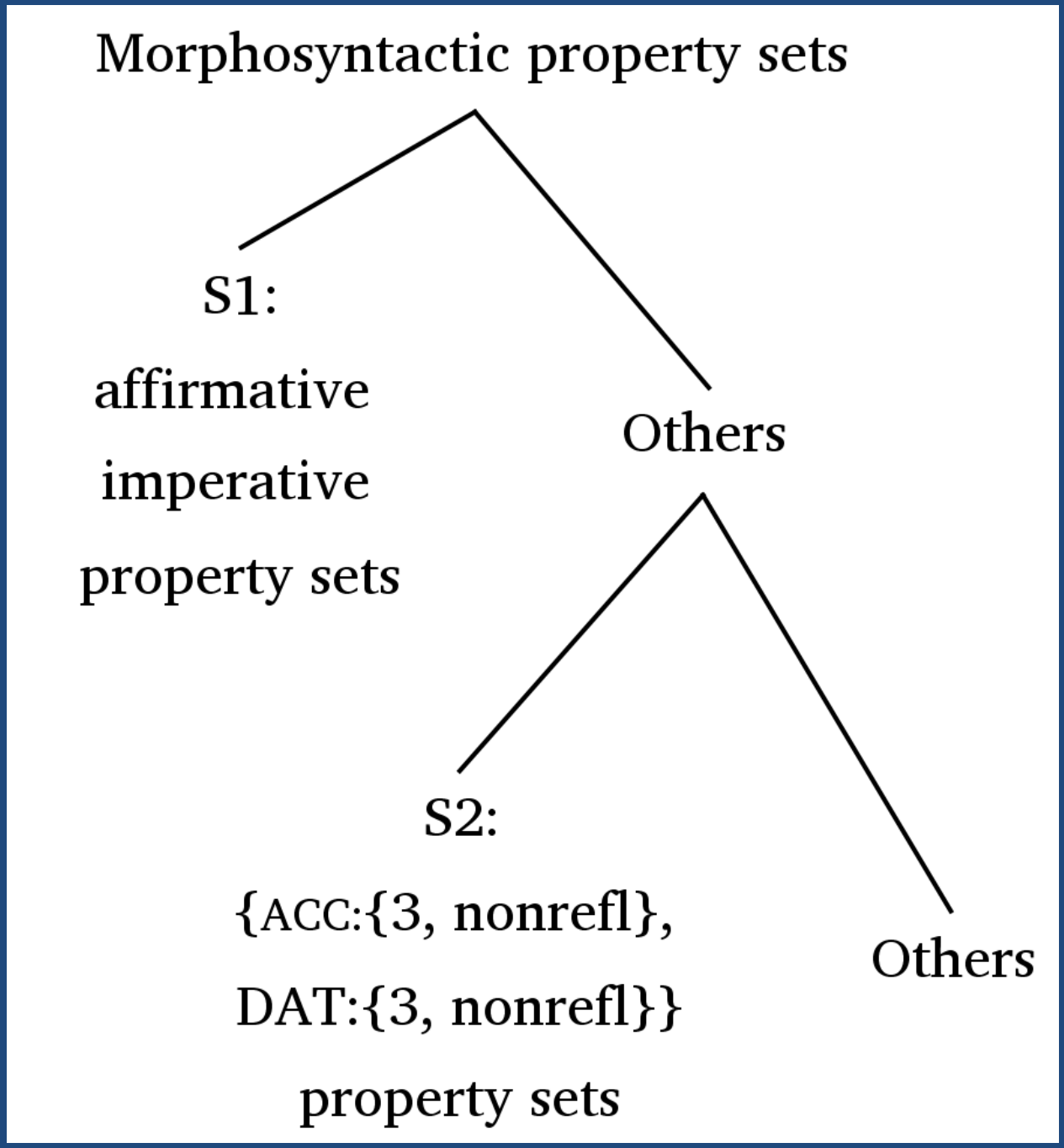
(41)

The set $\mathbf{S}_2$ of ***composition-inversion property sets*** for (40):

$\mathbf{S}_2$ is the smallest set containing every well-formed extension of (a) that is not an extension of (b):

a. {ACC:{3, nonrefl}, DAT:{3, nonrefl}}
b. {affirmative, imperative}.

(42)


Morphosyntactic property sets
S1:
affirmative
imperative
property sets
Others
S2:
{ACC:{3, nonrefl},
DAT:{3, nonrefl}}
property sets
Others

(43) Proof of *(Vous) me la présentez*

‘You introduce her to me’

(44) Proof of *(Vous) la leur présentez*

‘You introduce her to them’

(45) Proof of *Présentez-la-moi!*

‘Introduce her to me!’

(N.B.: Finally in a clitic group, *me, te* assume the shapes *moi, toi*.)

4.3 Predicting the "*me-lui* constraint"

(17) Basic template for proclitic local-argument pronouns

A (ACC or DAT)	B (ACC)	C (DAT)
me	*le*	*lui*
te	*la*	*leur*
se	*les*	
nous		
vous		

(46)

	A	B	C		
a.	*Jean*	*me*		*verra.*	‘Jean will see me.’
b.	*Jean*		*les*	*verra.*	‘Jean will see them.’
c.	*Jean*	*me*	*les*	*présentera.*	‘Jean will introduce them to me.’
d.	*Jean*		*les*	*lui présentera.*	‘Jean will introduce them to her.’
e.	**Jean*	*me*		*lui présentera.*	Putatively: ‘Jean will introduce me to her.’

(47) Person Case Constraint:
*1st/2nd/*se* acc. clitic + nonethical dat. clitic
(Rezac 2010:155)

(48) Czech

Chci *mu* *tě* *ukázat.*
I.want him.DAT you.SG.ACC show.INF
‘I want to show you (sg.) to him.’

(49) Polish

Wczoraj *mnie* *jemu* *przedstawiłeś.*
yesterday me.ACC him.DAT introduced.2SG.MASC
‘Yesterday you (sg. masc.) introduced me to him.’

(50) Feature cooccurrence restriction

If SL-2 or SL-[5] is nonempty,
then SL-4 is empty.

(Miller & Sag 1997: 597)

(51) Feature cooccurrence restriction

{CLTS ⟨…, x : {*obj*},…,{*dat*},…⟩}
$\Rightarrow$ x : {*3*,*nonrefl*}

(Bonami & Boyé 2007: 305)

(50) Feature cooccurrence restriction

If SL-2 or SL-[5] is nonempty,
then SL-4 is empty.

(Miller & Sag 1997: 597)

(51) Feature cooccurrence restriction

{CLTS ⟨…, x : {*obj*},…,{*dat*},…⟩}
⇒ x : {*3*,*nonrefl*}

(Bonami & Boyé 2007: 305)

Because clitics from Groups A and C are introduced by the same rule block, the proposed analysis entails the "*me-lui* constraint"

(38b) **Block II.** Where μ = ACC or DAT,

V,{μ:{1sg}}:	$\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{me}](\langle X, \sigma \rangle)$
V,{μ:{2sg}}:	$\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{te}](\langle X, \sigma \rangle)$
V,{μ:{3, refl}}:	$\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{se}](\langle X, \sigma \rangle)$
V,{μ:{1pl}}:	$\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{nous}](\langle X, \sigma \rangle)$
V,{μ:{2pl}}:	$\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{vous}](\langle X, \sigma \rangle)$
V,{DAT:{3sg, nonrefl}}:	$\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{lui}](\langle X, \sigma \rangle)$
V,{DAT:{3pl, nonrefl}}:	$\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{leur}](\langle X, \sigma \rangle)$

Because clitics from Groups A and C are introduced by the same rule block, the proposed analysis entails the "*me-lui* constraint"

(38b) **Block II.** Where μ = ACC or DAT,

V,{μ:{1sg}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, \textit{me}](\langle X, \sigma\rangle)$
V,{μ:{2sg}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, \textit{te}](\langle X, \sigma\rangle)$
V,{μ:{3, refl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, \textit{se}](\langle X, \sigma\rangle)$
V,{μ:{1pl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, \textit{nous}](\langle X, \sigma\rangle)$
V,{μ:{2pl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, \textit{vous}](\langle X, \sigma\rangle)$
V,{DAT:{3sg, nonrefl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, \textit{lui}](\langle X, \sigma\rangle)$
V,{DAT:{3pl, nonrefl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, \textit{leur}](\langle X, \sigma\rangle)$

(52) σ = {AGR:{3sg}, fut,
affirm, ACC:{1sg}, DAT:{3sg, nonrefl}}

(53) PF($\langle$*présentera*, $\sigma\rangle$)

(38b) **Block II.** Where μ = ACC or DAT,

V,{μ:{1sg}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, me](\langle X, \sigma\rangle)$
V,{μ:{2sg}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, te](\langle X, \sigma\rangle)$
V,{μ:{3, refl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, se](\langle X, \sigma\rangle)$
V,{μ:{1pl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, nous](\langle X, \sigma\rangle)$
V,{μ:{2pl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, vous](\langle X, \sigma\rangle)$
V,{DAT:{3sg, nonrefl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, lui](\langle X, \sigma\rangle)$
V,{DAT:{3pl, nonrefl}}:	$\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_1, leur](\langle X, \sigma\rangle)$

(52) σ = {AGR:{3sg}, fut,
affirm, ACC:{1sg}, DAT:{3sg, nonrefl}}

(53) PF(⟨*présentera*, σ⟩)

(38b) **Block II.** Where μ = ACC or DAT,

V,{μ:{1sg}}:	⟨X, σ⟩ → [<**S**$_1$, *me*](⟨X, σ⟩)
V,{μ:{2sg}}:	⟨X, σ⟩ → [<**S**$_1$, *te*](⟨X, σ⟩)
V,{μ:{3, refl}}:	⟨X, σ⟩ → [<**S**$_1$, *se*](⟨X, σ⟩)
V,{μ:{1pl}}:	⟨X, σ⟩ → [<**S**$_1$, *nous*](⟨X, σ⟩)
V,{μ:{2pl}}:	⟨X, σ⟩ → [<**S**$_1$, *vous*](⟨X, σ⟩)
V,{DAT:{3sg, nonrefl}}:	⟨X, σ⟩ → [<**S**$_1$, *lui*](⟨X, σ⟩)
V,{DAT:{3pl, nonrefl}}:	⟨X, σ⟩ → [<**S**$_1$, *leur*](⟨X, σ⟩)

(52) σ = {AGR:{3sg}, fut,
affirm, ACC:{1sg}, DAT:{3sg, nonrefl}}

(53) PF(⟨*présentera*, σ⟩)

(38b) **Block II.** Where μ = ACC or DAT,

V,{μ:{1sg}}:	⟨X, σ⟩ → [< **S**$_1$, *me*](⟨X, σ⟩)
V,{μ:{2sg}}:	⟨X, σ⟩ → [< **S**$_1$, *te*](⟨X, σ⟩)
V,{μ:{3, refl}}:	⟨X, σ⟩ → [< **S**$_1$, *se*](⟨X, σ⟩)
V,{μ:{1pl}}:	⟨X, σ⟩ → [< **S**$_1$, *nous*](⟨X, σ⟩)
V,{μ:{2pl}}:	⟨X, σ⟩ → [< **S**$_1$, *vous*](⟨X, σ⟩)
V,{DAT:{3sg, nonrefl}}:	⟨X, σ⟩ → [< **S**$_1$, *lui*](⟨X, σ⟩)
V,{DAT:{3pl, nonrefl}}:	⟨X, σ⟩ → [< **S**$_1$, *leur*](⟨X, σ⟩)

(52) σ = {AGR:{3sg}, fut,
affirm, ACC:{1sg}, DAT:{3sg, nonrefl}}

(53) PF(⟨*présentera*, σ⟩)

(38b) **Block II.** Where μ = ACC or DAT,

V,{μ:{1sg}}:	⟨X, σ⟩ → [<**S**$_1$, *me*](⟨X, σ⟩)
V,{μ:{2sg}}:	⟨X, σ⟩ → [<**S**$_1$, *te*](⟨X, σ⟩)
V,{μ:{3, refl}}:	⟨X, σ⟩ → [<**S**$_1$, *se*](⟨X, σ⟩)
V,{μ:{1pl}}:	⟨X, σ⟩ → [<**S**$_1$, *nous*](⟨X, σ⟩)
V,{μ:{2pl}}:	⟨X, σ⟩ → [<**S**$_1$, *vous*](⟨X, σ⟩)
V,{DAT:{3sg, nonrefl}}:	⟨X, σ⟩ → [<**S**$_1$, *lui*](⟨X, σ⟩)
V,{DAT:{3pl, nonrefl}}:	⟨X, σ⟩ → [<**S**$_1$, *leur*](⟨X, σ⟩)

(52) σ = {AGR:{3sg}, fut,
affirm, ACC:{1sg}, DAT:{3sg, nonrefl}}

(53) PF(⟨*présentera*, σ⟩)

(54) ***Narrowness***. Where (a) and (b) are realization rules, (a) is narrower than (b) iff either
(i) C = C′ and τ_2 is a proper subset of τ_1 or
(ii) C is a proper subset of C′.
(a) C, τ_1: $\langle X, \sigma\rangle \rightarrow f_1(\langle X, \sigma\rangle)$
(b) C′, τ_2: $\langle X, \sigma\rangle \rightarrow f_2(\langle X, \sigma\rangle)$

4.4 A unified account of proclitic and enclitic pronouns

(55) A hypothetical possibility

a. Paradigm function:
$PF(\langle X, \sigma\rangle) = [[\circ \mathbf{S}_3, \text{II}, \text{I}] : \langle X, \sigma\rangle]$

b. Block I: C, {}: $\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_3, y](\langle X, \sigma\rangle)$

c. Block II: C, {}: $\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_3, z](\langle X, \sigma\rangle)$

d. $\mathbf{S}_3 = \{\sigma_2\}$

(56) a. $PF(\langle X, \sigma_1\rangle) = \langle zyX, \sigma_1\rangle$

b. $PF(\langle X, \sigma_2\rangle) = \langle Xzy, \sigma_2\rangle$

(55) A hypothetical possibility

a. Paradigm function:

$PF(\langle X, \sigma \rangle) = [[\circ \mathbf{S}_3, \text{II}, \text{I}] : \langle X, \sigma \rangle]$

b. Block I: C, {}: $\langle X, \sigma \rangle \rightarrow [< \mathbf{S}_3, y](\langle X, \sigma \rangle)$

c. Block II: C, {}: $\langle X, \sigma \rangle \rightarrow [< \mathbf{S}_3, z](\langle X, \sigma \rangle)$

d. $\mathbf{S}_3 = \{\sigma_2\}$

(56) a. $PF(\langle X, \sigma_1 \rangle) = \langle zyX, \sigma_1 \rangle$

b. $PF(\langle X, \sigma_2 \rangle) = \langle Xzy, \sigma_2 \rangle$

(55) A hypothetical possibility

a. Paradigm function:
$PF(\langle X, \sigma \rangle) = [[\circ \mathbf{S}_3, \text{II}, \text{I}] : \langle X, \sigma \rangle]$

b. Block I: C, {}: $\langle X, \sigma \rangle \rightarrow [< \mathbf{S}_3, y](\langle X, \sigma \rangle)$

c. Block II: C, {}: $\langle X, \sigma \rangle \rightarrow [< \mathbf{S}_3, z](\langle X, \sigma \rangle)$

d. $\mathbf{S}_3 = \{\sigma_2\}$

(56) a. $PF(\langle X, \sigma_1 \rangle) = \langle zyX, \sigma_1 \rangle$

b. $PF(\langle X, \sigma_2 \rangle) = \langle Xzy, \sigma_2 \rangle$

(55) A hypothetical possibility

a. Paradigm function:
$PF(\langle X, \sigma \rangle) = [[\circ \mathbf{S}_3, \text{II}, \text{I}] : \langle X, \sigma \rangle]$

b. Block I: C, {}: $\langle X, \sigma \rangle \rightarrow [< \mathbf{S}_3, y](\langle X, \sigma \rangle)$

c. Block II: C, {}: $\langle X, \sigma \rangle \rightarrow [< \mathbf{S}_3, z](\langle X, \sigma \rangle)$

d. $\mathbf{S}_3 = \{\sigma_2\}$

(56) a. $PF(\langle X, \sigma_1 \rangle) = \langle zyX, \sigma_1 \rangle$

b. $PF(\langle X, \sigma_2 \rangle) = \langle Xzy, \sigma_2 \rangle$

(55) A hypothetical possibility

a. Paradigm function:
$PF(\langle X, \sigma\rangle) = [[\circ \mathbf{S}_3, \text{II}, \text{I}] : \langle X, \sigma\rangle]$

b. Block I: C, {}: $\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_3, y](\langle X, \sigma\rangle)$

c. Block II: C, {}: $\langle X, \sigma\rangle \rightarrow [<\mathbf{S}_3, z](\langle X, \sigma\rangle)$

d. $\mathbf{S}_3 = \{\sigma_2\}$

(56) a. $PF(\langle X, \sigma_1\rangle) = \langle zyX, \sigma_1\rangle$

b. $PF(\langle X, \sigma_2\rangle) = \langle Xzy, \sigma_2\rangle$

(55) A hypothetical possibility

a. Paradigm function:
$PF(\langle X, \sigma \rangle) = [[\circ \mathbf{S}_3, \text{II}, \text{I}] : \langle X, \sigma \rangle]$

b. Block I: C, {}: $\langle X, \sigma \rangle \rightarrow [< \mathbf{S}_3, y](\langle X, \sigma \rangle)$

c. Block II: C, {}: $\langle X, \sigma \rangle \rightarrow [< \mathbf{S}_3, z](\langle X, \sigma \rangle)$

d. $\mathbf{S}_3 = \{\sigma_2\}$

(56) a. $PF(\langle X, \sigma_1 \rangle) = \langle zyX, \sigma_1 \rangle$

b. $PF(\langle X, \sigma_2 \rangle) = \langle Xzy, \sigma_2 \rangle$

(57) a. *Me* *lo* *spedisce*.
1SG.DAT 3SG.ACC sends
‘He sends it to me.’

b. *Spedisci-me-lo!*
send.IMPV-1SG.DAT-3SG.ACC
‘Send it to me!’

(58)

a. *O* *Paulo* *deu-no-lo.*
the Paulo gave-1.PL.DAT-3.SG.MASC.ACC
‘Paulo gave it to us.’

b. *O* *Paulo* *não* *no-lo* *deu.*
the Paulo NEG 1.PL.DAT-3.SG.MASC.ACC gave
‘Paulo didn't give it to us.’

The proposed analysis does not require clitic sequences to be generated as a separate constituent that is positioned as a unit with respect to its host.

(59)

a. *Je leur y en ai porté.*
'I carried some there to them.'

b. *Envoyez-leur-y-en!*
'Send some to them there!'

(60)

Realization rules for the "adverbial" clitics

a. **Block EN.**

V, {EN:yes}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{en}](\langle X, \sigma \rangle)$

b. **Block Y.**

V, {Y:yes}: $\langle X, \sigma \rangle \rightarrow [<\mathbf{S}_1, \textit{y}](\langle X, \sigma \rangle)$

(61) Revised ***paradigm function***

PF($\langle$W, $\sigma\rangle$) =

[IV : [III : [[$\circ$**S**$_1$, [$\circ$**S**$_2$, II, I],[$\circ$**S**$_1$, Y, EN]] : $\langle$W, $\sigma\rangle$]]]

(61) Revised ***paradigm function***

PF(⟨W, σ⟩) =

[IV : [III : [[○$\mathbf{S}_1$, [○$\mathbf{S}_2$, II, I], [○$\mathbf{S}_1$, Y, EN]] : ⟨W, σ⟩]]]

(61) Revised ***paradigm function***

PF(⟨W, σ⟩) =

[IV : [III : [[○**S**$_1$, [○**S**$_2$, II, I], [○**S**$_1$, Y, EN]] : ⟨W, σ⟩]]]

(61) Revised ***paradigm function***

PF(⟨W, σ⟩) =

[IV : [III : [[○**S**$_1$, [○**S**$_2$, II, I],[○**S**$_1$, Y, EN]] : ⟨W, σ⟩]]]

(61) Revised ***paradigm function***

PF(⟨W, σ⟩) =

[IV : [III : [[○$\mathbf{S}_1$, [○$\mathbf{S}_2$, II, I],[○$\mathbf{S}_1$, Y, EN]] : ⟨W, σ⟩]]]

(17) Basic template for proclitic local-argument pronouns

A (ACC or DAT)	B (ACC)	C (DAT)
me	*le*	*lui*
te	*la*	*leur*
se	*les*	
nous		
vous		

(25) Basic template for enclitic local-argument pronouns

D (ACC)	E (ACC or DAT; 3rd person DAT only)
le	*moi*
la	*toi*
les	*lui*
	nous
	vous
	leur

(62) a. *Vous nous le donnez.*
'You give it to us.'

b. *Donnez-le-nous!* 'Give it to us!'

(17) Basic template for proclitic local-argument pronouns

A (ACC or DAT)	B (ACC)	C (DAT)
me	*le*	*lui*
te	*la*	*leur*
se	*les*	
nous		
vous		

(25) Basic template for enclitic local-argument pronouns

D (ACC)	E (ACC or DAT; 3rd person DAT only)
le	*moi*
la	*toi*
les	*lui*
	nous
	vous
	leur

(62) a. *Vous nous le donnez.*
'You give it to us.'

b. *Donnez-le-nous!* 'Give it to us!'

(59) a. *Je leur y en ai porté.*

'I carried some there to them.'

b. *Envoyez-leur-y-en!*

'Send some to them there!'

Proclitic and enclitic instances of the same clitic are introduced by the same rule, and the interaction of rule blocks conforms to a single structural generalization.

(61) Revised ***paradigm function***

PF(⟨W, σ⟩) =

[IV : [III : [[○$\mathbf{S}_1$, [○$\mathbf{S}_2$, II, I],[○$\mathbf{S}_1$, Y, EN]] : ⟨W, σ⟩]]]

Proclitic and enclitic instances of the same clitic are introduced by the same rule, and the interaction of rule blocks conforms to a single structural generalization.

(61) Revised ***paradigm function***

PF(⟨W, σ⟩) =

[IV : [III : [[○**S**$_1$, [○**S**$_2$, II, I],[○**S**$_1$, Y, EN]] : ⟨W, σ⟩]]]

The proposed analysis can be adapted to accommodate dialects other than the "standardized French" dialect considered here. Cf. e.g. (63), (64).

5. Two concluding remarks

(65) Huave completive affix *t*

a. [*mojk-o*]-*t*
face.down-THEME.VOWEL-COMPL
's/he lay face down'

b. *t*-[*e-mojk-o-r*]
COMPL-2-face.down-THEME.VOWEL-2.INTRANS
'you (sg.) lay face down'

(65) Huave completive affix *t*

a. [*mojk-o*]*-t*
face.down-THEME.VOWEL-COMPL
's/he lay face down'

b. *t*-[*e-mojk-o-r*]
COMPL-2-face.down-THEME.VOWEL-2.INTRANS
'you (sg.) lay face down'

(65) Huave completive affix *t*

a. [*mojk-o*]-*t*
face.down-THEME.VOWEL-COMPL
's/he lay face down'

b. *t*-[*e-mojk-o-r*]
COMPL-2-face.down-THEME.VOWEL-2.INTRANS
'you (sg.) lay face down'

(66)

a. A second default-suffixation operator $[>]_2$

Where σ is a morphosyntactic property set, X is an affix and Y is a stem:

$[>\mathbf{C}, X]_2$ is a function such that

$[>\mathbf{C}, X]_2(\langle Y, \sigma\rangle) = \langle XY, \sigma\rangle$ if Y satisfies **C**;

$= \langle YX, \sigma\rangle$ otherwise.

b. V, {completive}: $\langle X, \sigma\rangle \rightarrow$ $[>[\text{vowel}]\ldots[\text{cons}], t]_2(\langle X, \sigma\rangle)$

(66)

a. A second default-suffixation operator $[>]_2$

Where σ is a morphosyntactic property set, X is an affix and Y is a stem:
$[>\mathbf{C}, X]_2$ is a function such that
$[>\mathbf{C}, X]_2(\langle Y, \sigma\rangle) = \langle XY, \sigma\rangle$ if Y satisfies **C**;
$= \langle YX, \sigma\rangle$ otherwise.

b. V, {completive}: $\langle X, \sigma\rangle \rightarrow$
$[>[\text{vowel}]\ldots[\text{cons}], t]_2(\langle X, \sigma\rangle)$

(66)

a. A second default-suffixation operator $[>]_2$

Where σ is a morphosyntactic property set, X is an affix and Y is a stem:

$[>\mathbf{C}, X]_2$ is a function such that

$[>\mathbf{C}, X]_2(\langle Y, \sigma\rangle) = \langle XY, \sigma\rangle$ if Y satisfies **C**;

$= \langle YX, \sigma\rangle$ otherwise.

b. V, {completive}: $\langle X, \sigma\rangle \rightarrow$

$[>$[vowel]…[cons], $t]_2(\langle X, \sigma\rangle)$

(67) Lithuanian reflexive affix *si*

a. *laikaũ-si*
maintain.1SG-REFL
'I get along'

b. *iš-si-laikaũ*
PREVERB-REFL-maintain.1SG
'I hold my stand'

c. *su-si-pa-žìnti*
PREVERB_1-REFL-PREVERB_2-know.INF
'to become acquainted with'