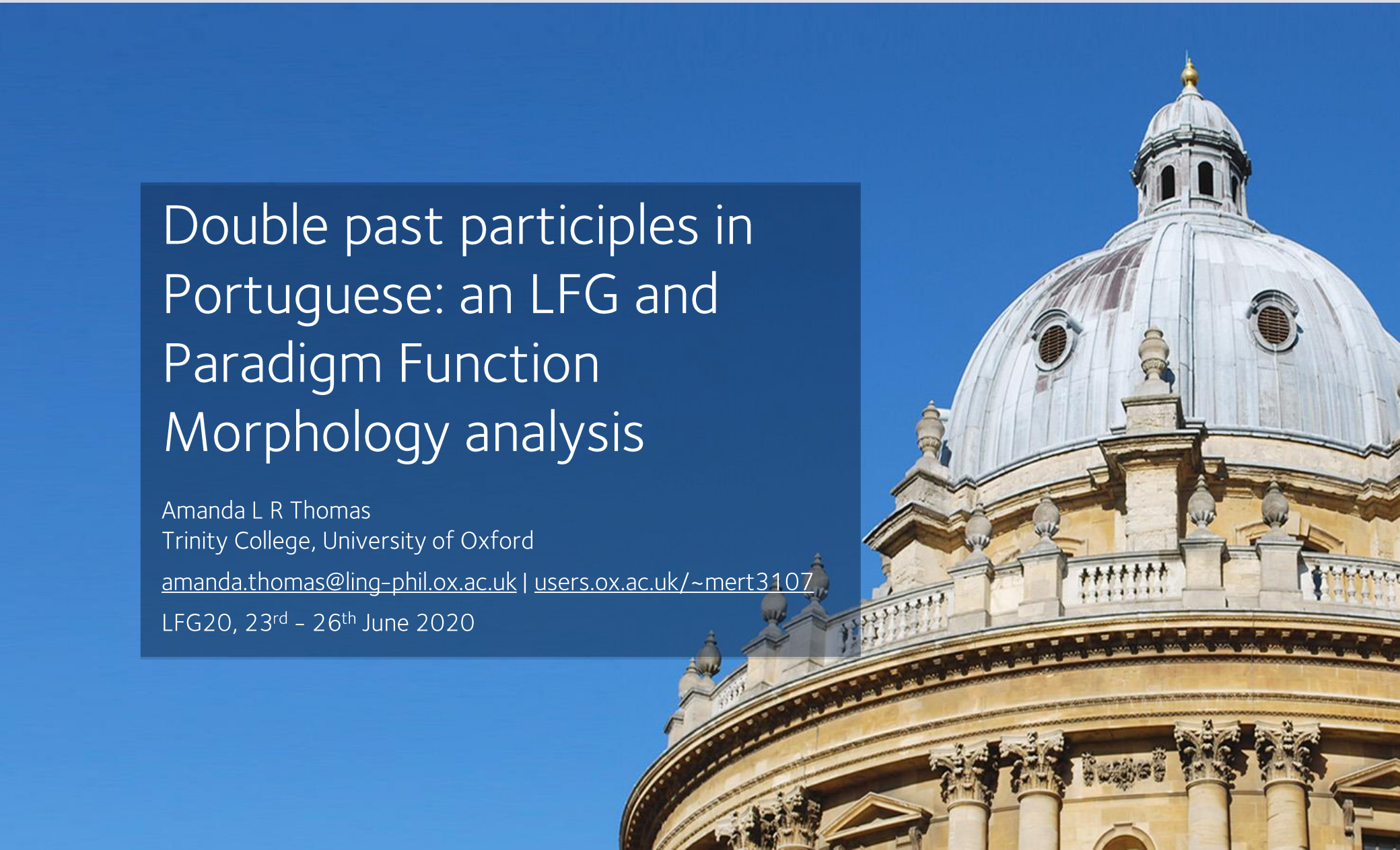
UNIVERSITY OF OXFORD



Double past participles in Portuguese: an LFG and Paradigm Function Morphology analysis

Amanda L R Thomas
Trinity College, University of Oxford
amanda.thomas@ling-phil.ox.ac.uk | users.ox.ac.uk/~mert3107

LFG20, 23rd – 26th June 2020

Roadmap

- Theory: combining existing LFG machinery with Paradigm Function Morphology, a powerful theory of inflectional morphology, to capture both morphology and syntax
- Case study: the ‘double past participle’ phenomenon in Portuguese
- Analysis: producing the correct morphological forms in the correct syntactic contexts, to give an analysis consistent with the LFG architecture

Theory: combining LFG and PFM

- The syntactic modules of LFG deal with fully inflected word-forms (Lexical Integrity): see for example Bresnan (2001), Sadler & Nordlinger (2018)
- Standard LFG does not have its own morphological component to produce these forms
- Dalrymple (2015) describes a syntax-morphology interface, assuming a separate module to provide inflectional morphology
- The inflectional morphology module, given an input of a lexeme and its morphosyntactic information, should provide a fully inflected word, paired with its morphosyntactic features
- Paradigm Function Morphology (Stump 2001) provides exactly this type of output, making it an ideal candidate for integration with LFG

Theory: basics of PFM

- PFM relates a lexeme to the inflected forms in its paradigm using the 'paradigm function' (PF)
- The input is a pairing of the root form (X) of a lexeme and a set of morphosyntactic features to be realised (σ): <X, σ>
- The output is an inflected form (Y) and the same set of features: <Y, σ>
- A simple example from English: plural of lexeme CAKE. σ = {num:PL}

Theory: more on PFM

- The paradigm function is made up of **rule blocks**, each containing a number of **unordered realisation rules**. These rules specify properties of the inflected form that they realise, e.g. concatenating phonetic material
- Within each block the narrowest applicable rule (that with the smallest number of possible inputs) applies to the input (**Panini's Principle**)
- If no rules in the block are applicable, the form stays the same: this is the **Identity Function Default**
- Example: rule block A contains two realisation rules. RR_1 applies to {num:SG, pers:3} and RR_2 applies to {num:SG}. RR_1 is *narrower* than RR_2 so applies to third person singular forms. RR_2 applies to all other singular forms
- Neither RR_1 nor RR_2 applies to plural forms, so the rule block does not alter plural form

Introducing double past participles 1

- In Romance and Germanic languages, the same form is generally used in active and passive periphrases:

(1) 'I had eaten the cake' vs. 'The cake was eaten'

- This is also the case for most Portuguese verbs, e.g. *comer* 'eat':

(2)

Tinha	*comido*	*muito*	*arroz*
have.IPFV.1SG	eat.PPL	much.MSG	rice.MSG

'I had eaten lots of rice'

(3)

O	*arroz*	*foi*	*comid-o*
the.MSG	rice.MSG	be.PRET.3SG	eat.PPL-MSG

'The rice was eaten'

Introducing double past participles 2

- In some Portuguese verbs, a different form is used in the two contexts: the **regular past participle** in the active (4) and an **irregular (strong) past participle** form in the passive (5):

(4) *A* *Maria* *tinha* *acendido* *as* *luzes*
the.FSG Maria.FSG have.IPFV.3SG light.PPL the.FPL lights.FPL
'Maria had turned on the lights' vs.

(5) *As* *luzes* *foram* *aces-as* *pela* *Maria*
the.FPL lights.FPL be.PRET.3PL light.PPL-FPL by.the.FSG Maria.FSG
'The lights were turned on by Maria' (Maiden 2013)

(Note: the past participle agrees in number and gender in passive contexts (3, 5), but does not agree in the active (2,4), but there are also differences in morphological form)

Implications of the ‘double past participle’ phenomenon

- Most Portuguese verbs use the same morphological form in the two syntactic contexts, but this is sometimes disrupted with arbitrary lexical variation, i.e. verb-to-verb variation
- The way the morphology relates to the syntax is not straightforward: an example of mismatch between the two modules of the grammar
- Unclear at what theoretical level the phenomenon should be captured: it is not purely morphological, like overabundance, where two forms with the same set of morphosyntactic features are in free variation
- Challenge: to capture the different facets of the phenomenon in morphology, syntax and the syntax-morphology interface

Producing possible forms for the past participle

- Using Taylor's (2012) treatment of the past participle in Portuguese as a complex stem, which is a stem itself formed on the basis of a simple stem, I set up PFM rules that give the possible **regular** and **irregular** (strong) past participle forms (see details in Appendix)

    1. Specify stem roots and theme vowels (vowels added to stem roots to form stems) for the relevant verbs

    2. Analyse different inflection classes

    3. Set up rules for stem formation and stem selection

- Then we can set up relationships between the two forms and the functions they fulfil

Worked example

Past participle used in the passive (pplPass) of *acender* 'to turn on': *aceso*

Verb is a member of inflection classes: **StrPPL**, (subclass) **S-StrPPL**, **C2** (verbs in *–er*), **DblPPL**

1. Stem selection rule 2 tells us that the pplPass is stem b + theme vowel (TV) [5]
2. To find out what stem b is, we look at the block of 'stem base selection rules'. The narrowest applicable rule is 7, which applies to **StrPPL** verbs
3. Stem b is a complex stem. 7 tells us that the base for stem b is stem h + TV[3]. Stem h is given in lexical data as "ace" and TV[3] is given as a null theme vowel.
4. "ace" is the input for the stem root formation rule 11 for stem b, which applies to verbs in the **S-StrPPL** class and adds an "s" → "aces"
5. We now finish 2 by adding TV[5], given in the lexical data as "o" → *aceso* ✓

Capturing lexical variation and morphosyntactic properties

- Challenge is to capture the mismatch between morphological and syntactic properties: **strong past participle forms are often (but not always) found in passive contexts (pplPass)**

Functions and forms	Active: pplAct	Passive: pplPass
Regular verb	Regular PPL →	Regular PPL
Irregular verb	Strong PPL ←	Strong PPL
Double past participle verb	Regular PPL	Strong PPL

- The distribution of forms and functions can be captured using rules of referral which link one verb form to another (example on next slide)
- Default rule of referral 3: the form used in the active context is also used in the passive: only for regular verbs
- Irregular verbs have the reverse rule of referral 4: the irregular form is used in active contexts, as well as passive contexts

Syntax-morphology interface: linking functions and forms

- For double past participle verbs, both regular and irregular forms are derived.
- Strong past participle forms are used with the morphosyntactic feature {vform: pplPass} and regular past participle forms with the feature {vform: pplAct}
- For regular verbs, the rule of the referral takes the form:

3 Where $\tau = \leftarrow$ {vform: pplAct} $\rightarrow$,
$SSR_{0,\tau}$, **V**$\cap$**Pt**($<\chi, \sigma>$) $=_{def}$ <[stema [4]], σ>
↓
$SSR_{0,\tau}$/{vform: pplPass},**V**$\cap$**Pt**($<\chi, \sigma>$) $=_{def}$ <[stema [4]], σ>

(7) regular verb *cantar* 'to sing': stem a = *cantad*, TV[4] = o
⇒ pplAct = *cantado* = pplPass

The syntax remains blind to purely morphological properties, and the morphological system works independently of syntax, consistent with LFG's modular architecture

UNIVERSITY OF OXFORD

Conclusions

- First LFG analysis of double past participles in Portuguese
- LFG and Paradigm Function Morphology can be combined to analyse phenomena with complex behaviour at the interface of morphology and syntax
- A solution for analysing mismatches between syntactic and morphological features, taking into account lexical variation, as exemplified by the Portuguese data
- Morphological theory can be integrated into the existing LFG architecture, providing a solution that can be applied to other morphosyntactic data

Acknowledgements and references

Bresnan, Joan (2001). Lexical functional syntax. Oxford: Blackwell.

Dalrymple, Mary (2015). 'Morphology in the LFG architecture.' In M Butt and T King, eds. *Proceedings of the LFG15 Conference*. Stanford: CSLI Publications.

Maiden, Martin (2013). 'The Latin third stem and its Romance descendants.' *Diachronica* 30 (4), 492–530.

Sadler, Louisa and Rachel Nordlinger (2018). 'Morphology in Lexical-Functional Grammar and Head-driven Phrase Structure Grammar'. In Jenny Audring & Francesca Masini (eds.), *The Oxford Handbook of Morphological Theory*. Oxford: OUP.

Stump, Gregory T (2001). *Inflectional morphology: a theory of paradigm structure*. Cambridge: CUP.

Taylor, Catherine (2012). *Formal relationships in the paradigm: a functional approach with a focus on the Romance verb*. PhD thesis, University of Essex.

I am grateful to the attendees of Oxford's Syntax Working Group who commented on an earlier version of this content. I also thank my supervisor, Dr Louise Mycock, for her advice and support.

This work was carried out while funded by an AHRC studentship.

Appendix: Example lexical data

Verb	Inflection class membership	stem root g	stem root h: used in pplPass	TV[1]	TV[2]	TV[3]	TV[4]: used in pplAct	TV[5]: used in pplPass	Regular past participle	Strong (irregular) past participle
CANTAR	C1	cant	—	a	a	Ø	o	o	cantado	—
COMER	C2	com	—	e	i	Ø	o	o	comido	—
PARTIR	C3	part	—	i	i	Ø	o	o	partido	—
ACEITAR	C1, DblPPL, StrPPL	aceit	—	a	a	Ø	o	e	aceitado	aceite
ENTREGAR	C1, DblPPL, StrPPL	entreg	—	a	a	Ø	o	e	entregado	entregue
GANHAR	C1, DblPPL, StrPPL	ganh	—	a	a	Ø	o	o	(ganhado)	ganho
GASTAR	C1, DblPPL, StrPPL	gast	—	a	a	Ø	o	o	(gastado)	gasto
LIMPAR	C1, DblPPL, StrPPL	limp	—	a	a	Ø	o	o	(limpado)	limpo
PAGAR	C1, DblPPL, StrPPL	pag	—	a	a	Ø	o	o	(pagado)	pago
ACENDER	C2, DblPPL, StrPPL, S-StrPPL	acend	ace	e	i	Ø	o	o	acendido	aceso
PRENDER	C2, DblPPL, StrPPL, S-StrPPL	prend	pre	e	i	Ø	o	o	prendido	preso
SUSPENDER	C2, DblPPL, StrPPL, S-StrPPL	suspend	suspen	e	i	Ø	o	o	suspendido	suspenso
MORRER	C2, DblPPL, StrPPL, T-StrPPL	morr	mor	e	i	Ø	o	o	morrido	morto
PÔR	C2, StrOnly, StrPPL, T-StrPPl	p	pos	e	i	Ø	o	o	—	posto
VER	C2, StrOnly, StrPPL, T-StrPPl	v	vis	e	i	Ø	o	o	—	visto
DIZER	C2, StrOnly, StrPPL, T-StrPPl	d	—	e	i	Ø	o	o	—	dito
ELEGER	C2, DblPPL, StrPPL, T-StrPPl	eleg	ele	e	i	Ø	o	o	elegido	eleito
ESCREVER	C2, StrOnly, StrPPL, T-StrPPl	escrev	escr	e	i	Ø	o	o	—	escrito
FAZER	C2, StrOnly, StrPPL, T-StrPPl	faz	fe	e	i	Ø	o	o	—	feito
EXPRIMIR	C3, DblPPL, StrPPL, S-StrPPL	exprim	expres	i	i	Ø	o	o	exprimido	expresso
ABRIR	C3, StrOnly, StrPPL, T-StrPPl	abr	aber	i	i	Ø	o	o	—	aberto
COBRIR	C3, StrOnly, StrPPL, T-StrPPl	cob	cober	i	i	Ø	o	o	—	coberto

StrPPL is a class of verbs consisting of the subclasses StrOnly and DblPPL. T-StrPPL and S-StrPPL are also subclasses of StrPPL, crosscutting StrOnly and DblPPL. Forms given in brackets are attested but not in current use.

Appendix: PFM rules for past participles

Stem selection rules (SSRs)

1. SSR0,{vform: pplAct},**V**∩**Pt**(<χ, σ>) =$_{def}$ <[stema [4]], σ>
2. SSR0,{vform: pplPass},**V**∩**Pt**(<χ, σ>) =$_{def}$ <[stemb [5]], σ>

Rules of referral

3. Where τ =← {vform: pplAct} →,
   SSR0,τ, **V**∩**Pt**(<χ, σ>) =$_{def}$ <[stema [4]], σ>
   ↓
   SSR0,τ/{vform: pplPass},**V**∩**Pt**(<χ, σ>) =$_{def}$ <[stema [4]], σ>
4. Where τ =← {vform: pplPass} →,
   SSR0,τ, **StrOnly**∩**Pt**(<χ, σ>) =$_{def}$ <[stemb [5]], σ>
   ↓
   SSR0,τ/{vform: pplAct},**StrOnly**∩**Pt**(<χ, σ>) =$_{def}$ <[stemb [5]], σ>
5. Where τ =← {vform: pplAct} →,
   SSR0,τ, **DblPPL**∩**Pt**(<χ, σ>) =$_{def}$ <[stema [4]], σ>
   ↓
   SSR0,τ/{vform: pplPass},**DblPPL**∩**Pt**(<χ, σ>) =$_{def}$ <[stemb [5]], σ>

Stem base selection rules (BSRs)

6. BSR0,{a,b},**V**∩**Pt**(<χ, ι>) =$_{def}$ <[stemg[2]], ι>
7. BSR0,{b},(**StrPPL**)∩**Pt**(<χ, ι>) =$_{def}$ <[stemh[3]], ι>
8. BSR0,{b},({dizer, eleger, escrever, fazer}∩**StrPPL**)∩**Pt**(<χ, ι>) =$_{def}$ <[stemh[2]], ι>

Stem root formation rules (SRs)

9. SR1,{a},V∩**Pt**(<χ, ι>) =$_{def}$ <[χi/d/], ι>for all i
10. SR1,{b},**T-StrPPL**∩**Pt**(<χ, ι>) =$_{def}$ <[χi/t/], ι> for all i
11. SR1,{b},**S-StrPPL**∩**Pt**(<χ, ι>) =$_{def}$ <[χi/s/], ι> for all i
12. SR1,{b},(**StrPPL**∩**C1**) ∩**Pt**(<χ, ι>) =$_{def}$ <[χi], ι>for all i

Reading the rules

{a,b} are complex stems
{g,h} are basic stems
Stem h reindexes to stem g if not otherwise given
Numbers in square brackets e.g. [2], [3] refer to theme vowels in the theme vowel array (partially exemplified in 'example lexical data')

Inflection classes

StrPPL: verbs with a strong (irregular) past participle
StrOnly: verbs with *only* a strong past participle
DblPPL: verbs with two past participle forms (regular and strong)
S-StrPPL: verbs with a strong past participle with /s/ e.g. *aceso* 'lit.PPL'
T-StrPPL: verbs with a strong past participle with /t/ e.g. *eleito* 'chosen.PPL'
C1: traditional inflection class 1, verbs in */ar/*
C2: traditional inflection class 1, verbs in */er/*
C3: traditional inflection class 1, verbs in */ir/*