

# UMA IMPLEMENTAÇÃO COMPUTACIONAL DE CONSTRUÇÕES VERBAIS PERIFRÁSTICAS EM FRANCÊS

Leonel Figueiredo de ALENCAR\*

- **RESUMO:** Este artigo descreve o tratamento da passiva e do passado composto na FrGramm, uma gramática computacional do francês implementada na Gramática Léxico-Funcional (LFG) usando o *software* XLE. Devido à dualidade de auxiliares e concordância do participípio passado (PTPST), a segunda perífrase exibe uma maior complexidade estrutural em francês do que em línguas como inglês e português, representando, conseqüentemente, um maior desafio à implementação computacional. Uma dificuldade adicional é a modelação das regularidades morfológicas e sintático-semânticas da passiva. A FrGramm resolve esse problema por meio de uma regra lexical produtiva. Também implementa as restrições que governam a formação das duas perífrases verbais, exceto a concordância do PTPST com o objeto direto. A implementação foi avaliada pela aplicação de um analisador sintático automático (*parser*) a 157 sentenças gramaticais e 279 construções agramaticais. Todas as sentenças do primeiro conjunto foram analisadas corretamente. Apenas duas construções do segundo que violam a precedência do auxiliar do passado composto sobre o da passiva foram analisadas como gramaticais. A FrGramm é a única gramática LFG do francês com essa cobertura atualmente disponibilizada livremente. Uma versão futura dará conta da concordância do PTPST com o objeto direto e evitará a hipergeração referida.
- **PALAVRAS-CHAVE:** Linguística computacional. Análise sintática automática profunda. Gramática léxico-funcional. LFG/XLE. Morfologia de estados finitos. Perífrases verbais em francês. Voz passiva.

## Introdução

A Gramática Léxico-Funcional (LFG, do inglês *Lexical-Functional Grammar*) é uma teoria gerativa amplamente difundida tanto em estudos teóricos, descritivos ou tipológicos quanto na linguística computacional. Esse modelo tem fundamentado análises de um expressivo número de línguas, das mais diferentes famílias linguísticas (BRESNAN, 2001). Muitas dessas análises foram implementadas no computador, em parte no âmbito de projetos de desenvolvimento de sistemas de análise sintática

---

\* Universidade Federal do Ceará (UFC), Fortaleza – CE - Brasil. Professor do Departamento de Letras Estrangeiras. leonel.de.alencar@ufc.br.

automática (*parsing*) de nível profundo, voltados para o processamento semântico de textos em linguagem natural.<sup>1</sup>

O francês, ainda nos anos de 1990, foi uma das primeiras línguas a terem suas estruturas sintáticas matematicamente descritas no formalismo da LFG e implementadas em sistemas de *parsing* (ZWEIGENBAUM, 1991; FRANK, 1996; SCHWARZE, 1998; BUTT et al., 1999). Por conta da evolução tecnológica, algumas dessas abordagens, como as de Frank (1996) e Schwarze (1998), acabaram se tornando obsoletas, pela impraticabilidade de executar os respectivos sistemas de *parsing* em plataformas atuais. Outro problema é os códigos-fonte dessas implementações ou os respectivos analisadores sintáticos (*parsers*) não estarem livremente disponíveis. Esse é também o caso das abordagens mais recentes, a saber, os *parsers* SxLFG e XLFG assim como as gramáticas do francês construídas para testá-los (CLÉMENT; KINYON, 2001; BOULLIER; SAGOT; CLÉMENT, 2005 ; CLÉMENT, 2014; SAGOT, [2015?]).

O presente artigo foca o tratamento de construções verbais perifrásticas na FrGramm, uma gramática computacional do francês que recentemente implementamos no *Xerox Linguistic Environment* (XLE), *software* que representa o estado da arte atual no desenvolvimento e *parsing* de gramáticas no formalismo da LFG (CROUCH et al., 2011).<sup>2</sup> Graças à amigabilidade e eficiência, esse sistema tem sido utilizado há mais de uma década tanto no ensino e pesquisa quanto em aplicações de escala industrial. Além disso, é distribuído gratuitamente, sob uma licença de uso para fins não comerciais.<sup>3</sup>

Uma alternativa ao XLE é o XLFG, mais voltado para o ensino e pesquisa em LFG (CLÉMENT; KINYON, 2001). No entanto, esse sistema não está disponível para *download*, mas deve ser utilizado *on-line* (CLÉMENT, 2014). Por outro lado, a gramática do francês mais completa implementada nesse sistema, disponível no respectivo *site*, tem cobertura muito limitada. Analisa perífrases com o auxiliar *avoir* ‘ter’, mas hipergera amplamente, implementando apenas ínfima parcela das restrições envolvidas nessas construções.

Como é comum na literatura, neste trabalho designamos por LFG/XLE a variante notacional do formalismo da LFG implementada no XLE. A motivação principal para o desenvolvimento da FrGramm em LFG/XLE foi disponibilizar livremente uma gramática de média cobertura sintática do francês para utilização no ensino e pesquisa em áreas como teoria gramatical formal, linguística computacional ou processamento automático de linguagem natural. Antes da FrGramm, a única gramática amplamente

---

<sup>1</sup> O levantamento mais atualizado e abrangente das línguas com gramáticas computacionais (ou fragmentos de gramática) implementadas no formalismo da LFG contém 27 línguas (MÜLLER, 2016, p. 213-214).

<sup>2</sup> Este artigo aprofunda aspectos de uma introdução à teoria da LFG e ao desenvolvimento de gramáticas computacionais no XLE, utilizando exemplos do francês, recentemente publicada em língua alemã, fruto de parceria com Christoph Schwarze (SCHWARZE; ALENCAR, 2016). A FrGramm é uma versão significativamente melhorada do fragmento de gramática do capítulo 8 desse livro. Na divisão de trabalho para elaboração dessa obra, coube ao autor deste artigo a implementação dos diferentes fragmentos de gramática e a redação das respectivas seções de capítulos. Essas gramáticas refletem intenso diálogo entre os dois autores. Por eventuais erros, contudo, este autor assume a inteira responsabilidade.

<sup>3</sup> Para obter o XLE, consultar a página <<http://www2.parc.com/isl/groups/nlft/xle/>>.

acessível para uso com o XLE, num contexto didático, era a gramática do inglês que integra a documentação desse sistema (KING, 2004). No entanto, como se sabe, o francês apresenta diversas particularidades sintáticas em relação ao inglês. Consequentemente, uma adaptação da referida gramática do inglês para processar o francês está longe de constituir uma tarefa trivial. Por outro lado, traduzir, do formalismo do XLFG para o do XLE, a gramática do francês de Clément (2014), acima referida, não seria tão difícil para um usuário experiente. O problema, porém, é que essa gramática, como apontamos, tem cobertura muito limitada e analisa como gramaticais exemplos simples que violam as regularidades da formação de tempos compostos em francês.

A FrGramm, que tem cobertura muito mais ampla, não padece desse tipo de hiper-geração. Entre outras abordagens, incorpora elementos de Frank (1996) e Schwarze (1998), as duas gramáticas computacionais do francês cujas implementações estão suficientemente detalhadas na literatura, mas não é uma reimplementação; pelo contrário, foi desenvolvida a partir do zero e preenche lacunas dessas duas propostas. É a única gramática do francês em LFG/XLE distribuída livremente na Internet, sob licença de uso que permite modificações pelos usuários e redistribuição das versões modificadas.<sup>4</sup> Desse modo, pode ser ampliada para cobrir outros fenômenos, adaptada para enfoques gramaticais diferentes ou para outras línguas.

Dentre os vários fenômenos implementados na FrGramm 1.0, a atual versão da gramática, sobressaem, pela maior complexidade e maior contraste com os fatos análogos em língua inglesa, as construções verbais perifrásticas do francês exemplificadas em (1)-(5). Essas perífrases constituem-se de uma forma finita (doravante VFIN) de *être* ‘ser’ ou *avoir* ‘ter’ e de um particípio (doravante PTCP).

- (1) *La fée est arrivée.*  
[a fada é chegada]  
‘A fada chegou.’
- (2) *La fée a dansé.*  
[a fada tem dançado]  
‘A fada dançou.’
- (3) *La fée est annoncée.*  
‘a fada é anunciada’
- (4) *La reine a forcé les chevaliers à achever la tâche.*  
[a rainha tem forçado os cavaleiros a concluir a tarefa]  
‘A rainha forçou os cavaleiros a concluir a tarefa.’
- (5) *Les chevaliers ont été forcés à achever la tâche.*  
[os cavaleiros têm sido forçados a concluir a tarefa]  
‘Os cavaleiros foram forçados a concluir a tarefa.’

---

<sup>4</sup> As condições de uso são detalhadas em <<http://creativecommons.org/licenses/by-nc-sa/4.0/>>. O código-fonte, os conjuntos-teste bem como os resultados da avaliação da gramática estão disponíveis em <https://github.com/lfg-french-grammar>.

As sentenças (1), (2) e (4) exemplificam o passado composto (*passé composé*). Diferentemente do português e do inglês, o francês, nesse tempo perifrástico, tal como o italiano e o alemão, exhibe uma divisão entre os intransitivos: verbos inacusativos como *arriver* ‘chegar’ selecionam o auxiliar *être*, enquanto verbos inergativos como *danser* selecionam *avoir* (FRANK, 1996). Um contraste adicional, representando fator extra de complexidade para uma implementação computacional, é a concordância exibida pelo particípio passado (doravante PTPST). No caso de verbos do primeiro grupo de intransitivos, o PTPST manifesta concordância com o sujeito (ver (1)); essa concordância é bloqueada nos verbos do segundo grupo (ver (2)). Por outro lado, o PTPST de verbos transitivos, em construções com a ordem canônica SVO, como em (4), não é flexionado; a flexão de gênero e número, porém, é obrigatória nesses verbos em construções com o objeto anteposto, como na oração relativa em (6):

(6) *Ils mangent les pêches que la reine a pelées.*

[eles comem os pêssegos que a rainha tem descascados]

‘Eles comem os pêssegos que a rainha descascou.’

A sentença (3) exemplifica a voz passiva num tempo simples (no caso, o presente do indicativo), enquanto (5) exemplifica a passiva no passado composto, reunindo as complexidades das duas construções perifrásticas.

A forma superficial de (3) é a mesma de (1). A exemplo de outras línguas românicas como o português, o particípio passivo (doravante PTPASS) exhibe flexão de gênero e número concordando com o sujeito, o que não ocorre em línguas como o inglês. Tal como (1), há uma analogia aparente de (3) com construções predicativas adjetivais do tipo de (7):

(7) *La dame est vaillante.*

‘A dama é valente.’

Na LFG, a passiva resulta da aplicação de uma regra lexical sobre as entradas de verbos que governam um objeto direto (doravante OBJ) (KAPLAN; BRESNAN, 1982). Essa regra modela as relações sistemáticas entre formas verbais ativas e passivas. Desse modo, essas últimas não precisam ser listadas no léxico, o que simplifica a codificação desse componente e representa uma grande economia de espaço de armazenamento. A implementação computacional dessa abordagem, contudo, não é trivial. De fato, é preciso dar conta tanto das alterações na morfologia e subcategorização verbais quanto das relações semânticas entre as duas diáteses.<sup>5</sup> Um fator de complicação são os verbos de controle do OBJ como *forcer* ‘forçar’. Nesses verbos, na ativa, o sujeito do infinitivo é controlado pelo OBJ da oração principal (ver (4)). Na passiva, porém, o controlador

---

<sup>5</sup> Diáteses são alternâncias regulares da valência verbal, compreendendo tanto fenômenos de voz, como na oposição entre ativa e passiva, quanto alternâncias não expressas por voz verbal (BUSSMANN, 2002).

passa a ser o sujeito da oração principal (ver (5)). Uma das principais vantagens do XLE é oferecer um eficiente mecanismo para implementação de regras lexicais. A utilização desse mecanismo numa determinada gramática, contudo, implica o preenchimento de duas condições: (i) formulação de restrições adequadas para dar conta dos exemplos gramaticais, excluindo, ao mesmo tempo, construções agramaticais que violam essas restrições; (ii) implementação de um analisador morfológico. Neste artigo, mostramos em que medida a FrGramm satisfaz essas duas exigências.

Pelos desafios que colocam, essas duas construções verbais perifrásticas são, portanto, bastante interessantes do ponto de vista do desenvolvimento de gramáticas computacionais. Essa é uma das razões de as termos escolhido como foco do presente artigo, que apresenta a FrGramm pela primeira vez ao público de língua portuguesa. Mostraremos como a FrGramm implementa esses fenômenos, de modo a analisar corretamente exemplos como (1)-(5) e com a construção análoga de (7), ao mesmo tempo excluindo exemplos agramaticais.

Uma implementação dessas perifrases é também relevante do ponto de vista teórico, haja vista as análises discrepantes de que têm sido objeto no quadro da LFG. Quais as propriedades computacionais de cada análise concorrente? Este artigo representa uma contribuição a essa linha de investigação, ao implementar uma dessas análises no XLE. Como a FrGramm é disponibilizada livremente, análises concorrentes poderão ser mais facilmente implementadas no mesmo sistema usando a FrGramm como base e comparadas em relação à complexidade, em termos dos recursos computacionais de tempo ou espaço consumidos (PRATT-HARTMANN, 2010).

Antes de concluir esta introdução, vejamos os principais pontos de divergência na análise das construções (1)-(5) e (7) no quadro atual da LFG. Patejuk e Przepiórkowski (2014), por exemplo, argumentam que, em polonês, na passiva, tal como na construção predicativa adjetival, o verbo *być* ‘ser’ é um verbo de alçamento, cujo único argumento semântico é um XCOMP.<sup>6</sup> Na LFG, essa é uma função gramatical com uma posição argumental aberta, a ser preenchida via controle funcional (BRESNAN, 2001). Desse modo, em sentenças como (3) ou (7), o sujeito sentencial realiza um argumento semântico não da cópula, mas do XCOMP. Conforme essa abordagem, subjaz a (3) e (7) uma estrutura bipredicacional: a primeira predicação é expressa pela cópula, a segunda, pelo XCOMP.

Essa abordagem, porém, não é consensual, como se pode constatar no ParGramBank, um *treebank* paralelo de 10 línguas, gerado por gramáticas LFG/XLE (SULGER et al., 2013). As divergências, nesse *corpus*, nas análises de línguas como o norueguês, inglês, alemão e polonês se referem ao estatuto do VFLEX e do PTCP, por um lado, e do adjetivo, por outro. A questão em relação ao adjetivo é se essa categoria instancia ou não um XCOMP. Nas análises do polonês, o PTCP, analogamente ao adjetivo de construções do tipo de (7), funciona como núcleo de um AP, o qual realiza o XCOMP

---

<sup>6</sup> O termo **verbo de alçamento** é empregado na literatura da LFG seguindo a tradição da gramática gerativa transformacional. No entanto, na análise desses verbos na LFG, não há movimento de constituinte, dado o caráter não transformacional dessa teoria.

do VFLEX. Nas análises do inglês, ao contrário, o PTCP é o verbo principal nas construções passivas, configurando uma estrutura monopredicacional, ao passo que o AP predicativo realiza a função gramatical fechada PREDLINK.

Nesses dois pontos, discrepam três das primeiras gramáticas computacionais do francês. Frank (1996), por exemplo, adota a análise bipredicacional para a passiva e os tempos compostos. O adjetivo predicativo realiza nessa abordagem a função ACOMP, que é um XCOMP adjetival. Schwarze (1998) e Butt et al. (1999), por sua vez, implementam uma análise monopredicacional, na qual o VFLEX é um auxiliar sem estrutura argumental. O adjetivo predicativo realiza em Schwarze (1998) um ACOMP, mas em Butt *et al.* (1999), a função fechada PREDLINK. A FrGramm implementa a análise monopredicacional para as construções verbais (1)-(5); na análise de (7), atribui ao AP a função XCOMP.

Além desta introdução, este trabalho se estrutura em mais 4 seções. Na próxima seção, delineamos o quadro teórico e o sistema computacional utilizados para implementar a FrGramm. A seção seguinte tem como objeto a metodologia, expondo os conjuntos de dados e os procedimentos empregados na implementação das perífrases verbais. Objeto da penúltima seção é a arquitetura geral da FrGramm, o papel de seus diferentes módulos, com ênfase na modelação das restrições envolvidas nas construções verbais perífrásticas, e a testagem da gramática. A última seção expõe as considerações finais.

## O modelo gerativo da LFG e o sistema XLE

A LFG é uma vertente da gramática gerativa (BRESNAN, 2001; FALK, 2001). Graças à formalização matematicamente explícita, gramáticas de línguas naturais elaboradas no formalismo da LFG são diretamente implementáveis no computador. A implementação computacional de fenômenos gramaticais oferece duas vantagens principais em relação a descrições formuladas numa língua natural e/ou não completamente formalizadas. A primeira é a possibilidade de utilização em aplicações de tecnologia da linguagem natural, como tradutores automáticos, programas de anotação de *corpora*, de extração de informações, de resolução de perguntas etc. O exemplo mais notável desse último tipo de aplicação é o IBM Watson, que em 2011 venceu dois campeões humanos no programa televisivo de perguntas e respostas norte-americano *Jeopardy* (BEST, 2013). Esse sistema se baseia no *parsing* sintático profundo por meio de formalismo análogo à LFG (MCCORD; MURDOCK; BOGURAEV, 2012). A segunda vantagem é a possibilidade de testar, de forma automática, a coerência interna e a adequação empírica das análises em vastos conjuntos de dados, como listas de sentenças gramaticais e agramaticais, *treebanks* etc.

Um atrativo da LFG para a comunidade acadêmica é a disponibilidade gratuita do XLE. Trata-se de ambiente de desenvolvimento e testagem de gramáticas bastante eficiente e amigável, que constrói automaticamente um *parser* para uma dada gramática elaborada na variante notacional do formalismo da LFG denominada LFG/XLE. Um

diferencial desse sistema em relação a alternativas como o XLFG é a possibilidade de integrar um transdutor lexical para análise morfológica, o que reduz significativamente o esforço de codificação do léxico. Outra vantagem do XLE é o suporte à implementação de geradores e tradutores automáticos.

Em LFG/XLE, uma gramática consiste minimamente de dois componentes, a saber, as regras de estrutura sintagmática anotadas e o léxico. Esse último pode constituir-se de (i) formas plenas e/ou (ii) lemas. No formato (i), há uma entrada lexical para cada forma flexionada. Em pequenas gramáticas, esse formato é mais fácil de implementar; em gramáticas de grande cobertura, porém, é inviável se a língua em questão não dispõe de um léxico de formas plenas que possa ser adaptado. O formato (ii) é o mais enxuto e de desenvolvimento menos custoso. Pressupõe, contudo, um componente morfológico implementado como um transdutor lexical, um tipo de autômato de estados finitos que associa formas flexionadas a representações lexicais (BEESLEY; KARTTUNEN, 2003). Mais adiante, veremos como o transdutor lexical que desenvolvemos no âmbito da FrGramm 1.0 simplifica enormemente a implementação computacional da passiva e do passado composto.

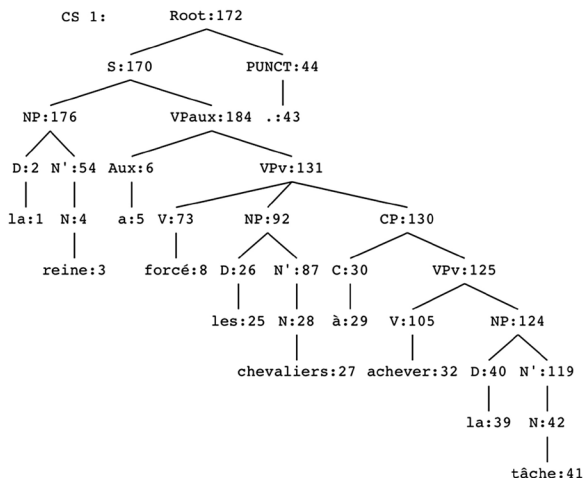
O *parser* gerado pelo XLE para uma dada gramática pode ser automaticamente aplicado a uma construção individual ou a um *corpus*. Para cada construção gramatical, o sistema gera automaticamente as diferentes representações sintáticas que a gramática atribui à construção. A **Figura 1** e a **Figura 2** exibem representações sintáticas produzidas pelo XLE.

Diferentemente dos modelos chomskyanos, como a Teoria da Regência e da Ligação (TRL) e o Programa Minimalista (PM), a LFG nega a existência, na linguagem humana, de transformações sintáticas (BRESNAN, 2001; FALK, 2001). Na LFG, as árvores, uma vez geradas, não sofrem mais modificações; apenas no léxico se admitem transformações. Desse modo, no caso de (4), a árvore gerada é apenas a da **Figura 1**. Nesse modelo, uma estrutura de constituintes (doravante estrutura C, do inglês *c-structure*) gramatical projeta um outro nível de representação, que é a estrutura funcional (estrutura F, do inglês *f-structure*). Na **Figura 2**, temos a estrutura F correspondente à estrutura C da **Figura 1**. Na **Figura 1**, *CS 1* no canto superior esquerdo indica que se trata da primeira estrutura C atribuída à sentença pelo *parser* (no caso, só há uma, pois se trata de sentença não ambígua). A projeção dos nós da estrutura C da **Figura 1** sobre a estrutura F da **Figura 2** é representada por meio dos índices numéricos dessas estruturas. Por exemplo, o nó mais alto da **Figura 1**, a categoria *Root* ('raiz' em inglês), que representa a sentença matriz, porta o índice 172. O nó S, que representa a sentença, é o nó 170, enquanto o VPaux (VP com auxiliar) é representado pelo índice 184. Os índices 172, 170 e 184 designam, na **Figura 2**, a estrutura F de toda a sentença.

Estruturas F de sintagmas, como as dos NPs ou a do VPaux da **Figura 1**, resultam da unificação das estruturas F dos seus constituintes. A unificação é a operação matemática fundamental da LFG e de modelos análogos, como a HPSG (MÜLLER, 2016). Essa operação reúne as informações de duas ou mais estruturas F em uma única estrutura,

desde que os valores dos diferentes atributos não conflitem (FRANCEZ; WINTNER, 2012, p. 85).

**Figura 1** – Estrutura C de (4) gerada pelo XLE a partir da FrGramm 1.0



Fonte: Elaboração própria.

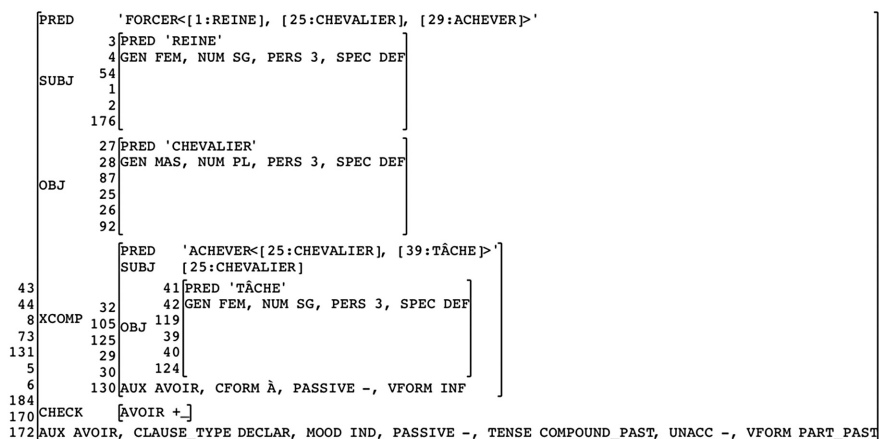
Estruturas F são matrizes de atributos e valores (AVMs, do inglês *attribute-value matrices*). Formalizam a noção de traço (*feature*), recorrente em diversas teorias linguísticas. Nesse contexto, um traço é um atributo (como GEN ‘gênero’ e NUM ‘número’ na **Figura 2**) com um valor (FEM ‘feminino’, SG ‘singular’ etc.). Por exemplo, conforme a **Figura 2**, *la reine* tem os traços GEN=FEM, NUM=SG, PERS=3 e SPEC=DEF. Os dois últimos especificam a pessoa gramatical e a especificação do sintagma, que, no caso, é definida. A estrutura F da sentença possui, entre outros, os traços *CLAUSE\_TYPE=DECLAR*, *MOOD=IND* e *PASSIVE=—*, indicando que se trata de sentença declarativa, indicativa e ativa.

Atributos do tipo de NUM ou PERS da **Figura 2** têm valores atômicos, que podem ser de três tipos: (i) uma cadeia de caracteres, como em NUM=SG, (ii) um número natural, como em PERS=3, ou (iii) um valor de verdade (“+” ou “—”), como em *PASSIVE=—*. Além disso, atributos podem ter valores não atômicos. O poder descritivo das AVMs como formalismo para a descrição de estruturas linguísticas advém justamente da possibilidade de um atributo ter como valor uma outra AVM. Com isso, esse formalismo pode dar conta da recursividade das estruturas sintáticas nas línguas naturais. Exemplos de atributos com um valor complexo na **Figura 2** são as funções gramaticais SUBJ (sujeito), OBJ e XCOMP.



**Figura 2** – Estrutura F de (4) gerada pelo XLE a partir da FrGramm 1.0

"La reine a forcé les chevaliers à achever la tâche."



Fonte: Elaboração própria.

Estruturas F não só codificam as propriedades gramaticais das sentenças, mas também constituem *input* para o processamento semântico (MÜLLER, 2016). A contribuição semântica de cada núcleo lexical individual para a construção do significado sentencial é representada por meio do atributo PRED. Núcleos funcionais (determinantes, auxiliares etc.) não possuem um atributo PRED, uma vez que sua contribuição para a estrutura F da sentença é meramente gramatical. No caso de núcleos lexicais avalentes, o valor de PRED, denominado forma semântica, é convencionalmente representado pelo lema entre aspas simples, por exemplo PRED='REINE'. No caso de núcleos lexicais portadores de valência, como o verbo *forcer* em (4), a forma semântica é denominada forma lexical e especifica a valência entre parênteses angulares (FALK, 2001, p. 13). Desse modo, a entrada lexical para uma forma ativa desse verbo contém a equação (8):

$$(8) \text{ PRED='FORCER'} < (^ \text{SUBJ}) (^ \text{OBJ}) (^ \text{XCOMP}) >'$$

Fórmulas como (8) são chamadas esquemas funcionais. Esses esquemas não só integram as entradas lexicais como também são utilizados como anotações nas regras de estruturação sintagmática para restringir a projeção da estrutura C sobre a estrutura F. Em (8), o verbo *forcer* é um predicado de três argumentos, a serem realizados, respectivamente, pelas estruturas F do SUBJ, OBJ e XCOMP. A realização de argumentos de um predicado é governada por dois princípios de boa formação de estruturas F: a Condição da Completude determina que todos os argumentos sejam realizados, enquanto a Condição da Coerência exclui as funções gramaticais regidas não especificadas na valência do predicado.

Na **Figura 2**, a fórmula geral (8) é instanciada como (9):

(9) PRED='FORCER<[1:REINE], [25:CHEVALIER], [29:ACHEVER]>'

Em (9), os lugares vazios do predicado de (8) estão preenchidos pelas estruturas F do SUBJ, OBJ e XCOMP, referidas, respectivamente, pelos índices 1, 25 e 29. Lugares argumentais em formas semânticas só podem ser preenchidos por estruturas F que possuem um atributo PRED. Por conveniência mnemônica, o XLE também insere, nos lugares argumentais de formas semânticas de valência saturada, as representações ortográficas dos predicados das funções gramaticais que realizam esses argumentos, como, no caso de (9), por exemplo, o lema REINE na primeira posição argumental.

O algoritmo de *parsing* deriva, para a sentença (3), a fórmula (9) a partir de (8) por meio das anotações funcionais.<sup>7</sup> O símbolo “^” corresponde à metavariable “^” na notação tradicional da LFG. Essa metavariable é instanciada, na estrutura F de um constituinte, por variável que designa a estrutura F do nó pai dessa categoria. No exemplo em tela, “^” refere-se ao nó pai de *forcé*, ou seja, a categoria V. Por funcionar como núcleo do VPaux, as informações associadas a V são projetadas para a estrutura F da sentença. Desse modo, uma expressão do tipo (^ GF) na forma lexical de um verbo, onde GF designa uma função gramatical, equivale a ‘GF da sentença’, por exemplo, (^ OBJ) equivale a ‘objeto direto da sentença’.

Para concluir esta seção, tratamos das regras lexicais, que desempenham um papel fundamental na análise das diáteses na LFG. Essas regras, junto ao restante aparato formal da teoria, dispensam a postulação de transformações sintáticas. Equivalem a funções que, aplicadas sobre entradas lexicais, geram outras entradas. No XLE, essas operações manipulam apenas os esquemas funcionais das entradas; não podem, portanto, manipular a forma dos lexemas, derivando uma forma passiva como *forçados* a partir da sufixação da forma ativa. O XLE, contudo, permite contornar essa deficiência, integrando um analisador morfológico ao *parser*, solução adotada pela FrGramm 1.0, como veremos mais adiante.

Agora, nos limitamos a uma formalização simplificada da regra da passiva em línguas do tipo do francês. Tarefa dessa regra é modelar a relação sistemática entre os verbos principais de exemplos como (3) e (10), por um lado, e (11), por outro. São os seguintes os fatos principais a serem modelados: (i) a todo particípio passivo corresponde uma forma ativa que rege um OBJ; (ii) o OBJ da forma ativa é realizado como SUBJ na forma passiva; (iii) o SUBJ da ativa é realizado facultativamente como oblíquo na passiva (SCHWARZE; ALENCAR, 2016, p. 149).

(10) *La fée est annoncée par le chevalier.*

‘A fada é anunciada pelo cavaleiro.’

(11) *Le chevalier annonce la fée.*

‘O cavaleiro anuncia a fada.’

---

<sup>7</sup> Um detalhamento do algoritmo de *parsing* da LFG foge ao escopo do presente trabalho. Consulte-se a esse respeito, por exemplo, Bresnan (2001, p. 56-60).

- (12) (i) forma ativa: ‘ANNONCER< (^ SUBJ) (^ OBJ)>’  
 (ii) 1ª forma passiva: ‘ANNONCER< (^ OBL) (^ SUBJ)>’  
 (iii) 2ª forma passiva: ‘ANNONCER< NULL (^ SUBJ)>’  
 (iv) grade temática: AGENTE TEMA

Essas generalizações estão sintetizadas em (12) (SCHWARZE; ALENCAR, 2016). Em (12) (ii), a função gramatical OBL, em línguas como o francês, é a dos complementos verbais preposicionados não passíveis de pronominalização por clítico dativo, diferindo, portanto, da função OBJ2 (objeto indireto ou objeto secundário), passível dessa cliticização. Nesse caso, OBL expressa o agente da passiva. Em (12) (iii), NULL representa a não realização desse argumento. O nível (iv) é modelado na LFG como estrutura A (*a-structure*, do inglês *argument structure*), desempenhando um importante papel na arquitetura da teoria (FALK, 2001, p. 105 *et passim*). Essa estrutura, porém, não está implementada no XLE.

Tomando como base apenas as propriedades de (12), a passiva pode ser modelada como uma operação que incide sobre as funções gramaticais da entrada lexical da forma ativa, derivando, por meio das transformações de (13), duas entradas lexicais para o participio passivo. A primeira entrada subjaz a exemplos como (10), a segunda, a exemplos como (3).

- (13) {SUBJ → OBL | SUBJ → NULL}  
 OBJ → SUBJ

Na primeira linha de (13), temos uma disjunção lógica, expressa pelo conector “|”. Essa primeira parte da regra compreende duas alternativas: pela primeira, o SUBJ é convertido em OBL; pela segunda, o SUBJ é convertido em NULL, o que resulta no seu apagamento. Na segunda linha da regra é codificada a transformação do OBJ em SUBJ.

## Dados e procedimentos

Esta seção trata dos dois conjuntos de dados utilizados na implementação da passiva e do passado composto na FrGramm 1.0. O conjunto-teste positivo define o recorte gramatical implementado. O conjunto-teste negativo permite verificar se as restrições que caracterizam os fenômenos em questão foram corretamente implementadas, evitando a hipergeração. Objeto da seção são também as noções de fragmento e de desenvolvimento em espiral.

Como vimos, a LFG é um modelo matematicamente explícito. Disso decorre que a modelação de um fenômeno gramatical precisa se restringir a um fragmento da língua, i.e. um conjunto definido de sentenças. O trabalho com fragmentos é praxe na sintaxe computacional (FRANCEZ; WINTNER, 2012).

Estreitamente relacionada a essa prática é a adoção de um desenvolvimento em espiral. Conforme essa técnica de *design* de *software*, desenvolve-se primeiro uma versão mais simples de um programa (um protótipo), que abarca apenas parte do problema que o *software* objetiva resolver; em seguida, em sucessivas etapas, esse protótipo é progressivamente incrementado, de modo a dar conta de cada vez mais facetas do problema (ZELLE, 2004). A aplicação dessa técnica na elaboração de uma gramática consiste em começar com a implementação de um fragmento reduzido do recorte gramatical, expandindo a cobertura desse protótipo por meio da implementação de fragmentos sucessivamente mais abrangentes.

Para poder ser testado no computador, um modelo linguístico computacional deve constituir um fragmento de gramática capaz de analisar construções que exemplificam as diferentes facetas do fenômeno em questão. Isso implica implementar outros fenômenos presentes nessas construções. Por exemplo, um fragmento de gramática capaz de analisar sentenças passivas precisa tratar também da concordância, da ordem das palavras, da estrutura do sintagma posicional etc.

A LFG concebe um fenômeno gramatical como uma série de restrições que definem um conjunto de construções gramaticais por oposição a um conjunto de construções agramaticais. Essa concepção tem duas consequências imediatas para a implementação computacional de uma análise. A primeira é que ela deve ser testada em relação a dois conjuntos-teste: o conjunto-teste positivo, com as sentenças gramaticais, e o conjunto-teste negativo, com construções que violam as restrições postuladas. A segunda consequência é que a implementação deve cobrir construções superficialmente análogas, mas fundamentalmente diferentes em termos de restrições, como, por exemplo, em (1), (3) e (7).

Nesses exemplos, temos uma mesma estrutura superficial, que podemos esquematizar como *SUBJ est X ‘SUBJ é X’*, onde *X* é um constituinte que concorda em gênero e número com o *SUBJ*. Trata-se, porém, de três construções distintas: (7) é uma construção predicativa adjetival, (1), uma sentença no passado composto, enquanto (3) é uma sentença passiva. Que restrições caracterizam a passiva, distinguindo-a das outras duas construções? É evidente que somente uma implementação conjunta das três construções permite estabelecer os conjuntos de restrições que as distinguem entre si.

O conjunto-teste positivo contém apenas sentenças construídas. A razão para não se utilizar, na construção de um fragmento de gramática, exemplos extraídos de textos reais é evitar uma série de dificuldades. Em primeiro lugar, para testar o fragmento em exemplos reais, seria preciso implementar um vasto léxico. Na fase inicial de desenvolvimento de um fragmento de gramática, isso significaria desviar esforços da complexa tarefa de modelação formal e implementação computacional da sintaxe. Em segundo lugar, exemplos reais de um determinado fenômeno normalmente instanciam

complexidades sintáticas que não se referem especificamente a esse fenômeno, como na ocorrência de construção passiva em (14).

- (14) [...] *cette date ou l'indication de l'endroit où elle se trouve est annoncée par la mention " à utiliser de préférence avant fin ..." ou par le symbole d'un sablier.*<sup>8</sup>  
[esta data ou a indicação do lugar onde ela se encontra é anunciada pela menção "a utilizar de preferência antes do fim de..." ou pelo símbolo de uma ampulheta.]

Dada a complexidade da passiva e do passado composto em francês, nos restringimos, para implementar esses fenômenos na FrGramm 1.0, ao recorte gramatical exemplificado em (1)-(5), (7), (10), (11) e (15)-(21). Esse recorte inclui, portanto, também a construção predicativa adjetival.

- (15) *La reine prie la dame de chanter dans les anciens châteaux blancs habités par des fées.*  
'A rainha solicita a dama a cantar nos antigos castelos brancos habitados por fadas.'
- (16) *Les dames ont été priées de danser.*  
'As damas foram solicitadas a dançar.'
- (17) *Le chevalier a été forcé à inviter les dames à danser.*  
'O cavaleiro foi forçado a convidar as damas a dançar.'
- (18) *La reine a ordonné aux chevaliers de danser.*  
'A rainha ordenou aos cavaleiros dançar.'
- (19) *La dame a été aimable.*  
'A dama foi amável.'
- (20) *La fée demande à être invitée à danser.*  
'A fada pede para ser convidada a dançar.'
- (21) *Les chevaliers ont été forcés par la reine à achever la tâche.*  
'Os cavaleiros foram forçados pela rainha a concluir a tarefa.'

Explicitamos em (22) o leque de fenômenos modelados na FrGramm 1.0. relacionados diretamente com a passiva e o passado composto. A concordância do PTPST com o OBJ não foi contemplada nessa versão (ver (6)).

- (22) (i) voz ativa e voz passiva de diferentes classes valenciais, incluindo verbos de controle do OBJ como *forcer* 'forçar', *prier* 'pedir' etc.  
(ii) tempos simples e passado composto na ativa  
(iii) passiva no presente e no passado composto  
(iv) particípio passivo como verbo principal e como adjunto adnominal  
(v) concordância nominal e verbal  
(vi) seleção do auxiliar do passado composto

---

<sup>8</sup> Exemplo extraído em 25/01/2016, por meio do Google, da URL <[http://ansm.sante.fr/Activites/Surveillance-du-marche-des-produits-cosmetiques/Periode-apres-ouverture-PAO/\(offset\)/1](http://ansm.sante.fr/Activites/Surveillance-du-marche-des-produits-cosmetiques/Periode-apres-ouverture-PAO/(offset)/1)>.

O conjunto-teste positivo que serviu de base para a implementação da FrGramm 1.0 consiste de 157 sentenças gramaticais. O conjunto-teste negativo, que contém 279 sentenças agramaticais, foi construído manualmente a partir do primeiro por meio da transformação sistemática das sentenças gramaticais em agramaticais. Por exemplo, a partir de (1), (16), (18) e (19) geraram-se sentenças agramaticais como (23)-(33), violando-se uma ou mais de uma das restrições relativas à concordância, forma verbal, seleção do auxiliar, passivização etc.

(23) \**La fée a arrivé.*

(24) \**La fée a arrivée.*

(25) \**La fée est arrivé.*

(26) \**La fée arrivée est.*

(27) \**La fée est est arrivée.*

(28) \**La fée est arriver.*

(29) \**La dame été a aimable.*

(30) \**La reine a ordonnée aux chevaliers de danser.*

(31) \**Les chevaliers sont ordonnés de danser.*

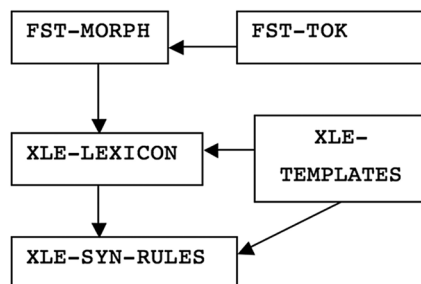
(32) \**La reine est ordonnée aux chevaliers de danser.*

(33) \**Les dames été ont priées de danser.*

### Aspectos da implementação

À FrGramm 1.0 subjazem os módulos da **Figura 3**, que configuram uma arquitetura padrão para gramáticas LFG/XLE (BUTT et al., 1999; KING, 2004; CROUCH et al., 2011): (i) FST-TOK, um toquenizador; (ii) FST-MORPH, um analisador morfológico; (iii) XLE-LEXICON, um conjunto de entradas lexicais; (iv) XLE-TEMPLATES, moldes análogos a macros parametrizadas de determinadas linguagens de programação; (v) XLE-SYN-RULES, regras independentes de contexto anotadas com esquemas funcionais. Os módulos com o prefixo FST são transdutores de estados finitos, implementados por meio do XFST (BEESLEY; KARTTUNEN, 2003). O prefixo XLE indica a implementação do componente em LFG/XLE.

**Figura 3** – Arquitetura da FrGramm 1.0



Fonte: Elaboração própria.

A partir desses componentes, o XLE constrói um *parser*, o qual pode ser aplicado na análise de sentenças inteiras ou sintagmas individuais. Para cada construção gramatical (conforme a gramática subjacente, no caso a FrGramm), o XLE apresenta o respectivo conjunto de estruturas C. Sentenças tratadas como sintaticamente ambíguas pela FrGramm, como (15), projetam mais de uma estrutura C. No caso em tela, são geradas pelo XLE duas estruturas C, comparem-se (34a) e (34b). Cada estrutura C válida, por sua vez, projeta uma ou mais de uma estrutura F, representando as diferentes leituras da sentença em termos funcionais.

- (34) a. *La reine* [<sub>VP</sub> [<sub>V</sub> *prie*] [<sub>NP</sub> *la dame*] [<sub>CP</sub> *de chanter*] [<sub>pp</sub> *dans les anciens châteaux ...*]].  
b. *La reine* [<sub>VP</sub> [<sub>V</sub> *prie*] [<sub>NP</sub> *la dame*] [<sub>CP</sub> *de chanter dans les anciens châteaux ...*]].

Detalhemos cada um dos componentes da **Figura 3**, começando pelo mais básico, o toquenizador FST-TOK. A função desse módulo é segmentar uma cadeia de caracteres, dada como entrada do *parser*, em uma sequência de *tokens*, i.e. palavras e sinais de pontuação, que são delimitados pelo símbolo “@”, como na toquenização da sentença (11) em (35). Além disso, realiza outra tarefa importante no pré-processamento de sentenças, que é a normalização (PALMER, 2010). Essa tarefa consiste em converter as formas variantes de um *token* em uma forma padrão, como no exemplo (35), em que o determinante *Le* ‘o’ é minúsculizado.

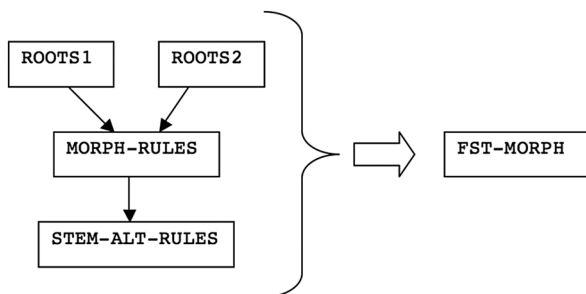
- (35) le@chevalier@annonce@la@fée@.@

O componente morfológico FST-MORPH é um transdutor lexical (BEESLEY; KARTTUNEN, 2003). Na versão atual da FrGramm, esse analisador se restringe a 39 verbos da 1ª conjugação, totalizando 1794 formas. Por exemplo, para a forma flexionada *annonce*, que instancia a 1ª e 3ª pessoas do singular do presente do indicativo e do subjuntivo do verbo *annoncer* ‘anunciar’, o transdutor retorna as quatro representações de (36).

- (36) annoncer+V+SBJP+3+SG  
annoncer+V+SBJP+1+SG  
annoncer+V+PRS+3+SG  
annoncer+V+PRS+1+SG

A exemplo das representações de (36), as análises geradas pelo componente morfológico constituem-se do lema seguido de uma sequência de etiquetas que codificam, respectivamente, a categoria lexical (V=verbo), o tempo e o modo (PRS=presente do indicativo, SBJP=presente do subjuntivo), a pessoa e o número (SG=singular).

**Figura 4** – Arquitetura do componente morfológico da FrGramm 1.0



Fonte: Elaboração própria.

O analisador morfológico FST-MORPH foi implementado conforme a arquitetura esquematizada na **Figura 4**. Os quatro componentes da esquerda foram compilados em transdutores e combinados por meio de operações de estados finitos para produzir o componente da direita. Os módulos ROOTS1 e ROOTS2 contêm radicais.<sup>9</sup> O primeiro consiste de radicais de verbos regulares como *regarder* ‘ver’ que não sofrem alterações na conjugação, ao passo que o segundo constitui-se de radicais de verbos como *annoncer* que sofrem algum tipo de alteração ortográfica sistemática. No caso de *annoncer*, <c> final é substituído por <ç> antes de vogal posterior, como em *annonçons* ‘anunciamos’. Quatro outros tipos de alteração sistemática no radical de verbos da 1ª conjugação foram contemplados. Os verbos *acheter* ‘comprar’, *répéter* ‘repetir’, *jeter* ‘jogar’ e *exiger* ‘exigir’ exemplificam esses quatro tipos.<sup>10</sup>

O componente MORPH-RULES é uma gramática no formalismo LEXC (BEESLEY; KARTTUNEN, 2003). Essa gramática modela a combinatória de radicais e flexões verbais. É compilada num transdutor que codifica uma relação de pares  $(p, w)$ , onde  $p$  é uma representação lexical do tipo de (36) e  $w$ , uma forma flexionada intermediária

<sup>9</sup> A atual versão do componente morfológico da FrGramm não contempla a morfologia derivacional, pelo que os componentes ROOTS1 e ROOTS2 contêm apenas raízes verbais.

<sup>10</sup> Por falta de espaço, não podemos detalhar mais esse aspecto. A construção do analisador morfológico será tema de trabalho futuro.



como, por exemplo, *mang<sup>^</sup>ons*, do paradigma de *manger* ‘comer’. O último componente da morfologia é STEM-ALT-RULES, que consiste de regras de reescrita que modelam as alternâncias ortográficas das cinco subclasses de verbos acima referidas. Essas regras se aplicam sobre formas intermediárias como *mang<sup>^</sup>ons* e derivam formas finais como *mangeons* ‘comemos’.

Como nosso foco é a implementação de construções verbais perifrásticas, vejamos como o analisador morfológico trata o particípio, forma verbal constitutiva dessas construções. Em (37), transcrevemos parte de sessão de linha de comandos do XFST. Por meio do comando `load`, carregamos o analisador (armazenado no arquivo binário `fst-morph`) e, em seguida, por meio do comando `up`, o aplicamos na análise de alguns particípios franceses.

```
(37) xfst[0]: load fst-morph
      Opening input file 'fst-morph'
      June 04, 2015 14:43:26 GMT
      Closing input file 'fst-morph'
      xfst[1]: up arrivé
      arriver+V+PTPST+UNERG
      arriver+V+PTPST+UNACC+M+SG
      arriver+V+PTPASS+M+SG
      xfst[1]: up dansé
      danser+V+PTPST+UNERG
      danser+V+PTPST+UNACC+M+SG
      danser+V+PTPASS+M+SG
      xfst[1]: up dansée
      danser+V+PTPST+UNACC+F+SG
      danser+V+PTPASS+F+SG
```

No FST-MORPH, os particípios do francês são classificados conforme o **Quadro 1**. Nessa classificação, a primeira divisão é entre particípios ativos (PTPST) e particípios passivos (PTPASS). O segundo critério é a concordância, que se aplica apenas na primeira subcategoria. Ao passo que o PTPASS sempre concorda com o seu SUBJ, o PTPST realiza essa concordância apenas com verbos de auxiliar *être*, como no caso de *arriver*. Tipicamente, são inacusativos (UNACC é abreviatura de *unaccusative*), enquanto os intransitivos que selecionam *avoir* (como *danser*) são inergativos (UNERG é abreviatura de *unergative*).<sup>11</sup>

---

<sup>11</sup> A distinção entre inacusativos e inergativos não corresponde exatamente à distinção entre verbos que selecionam *être* ou *avoir* no passado composto. Há várias exceções importantes (SCHWARZE; ALENCAR, 2016, p. 160).

**Quadro 1** – Classificação dos participios no FST-MORPH

DIÁTESE	CONCORDÂNCIA	
	COM	SEM
ATIVA	PTPST+UNACC	PTPST+UNERG
PASSIVA	PTPASS	∅

Fonte: Elaboração própria.

Os exemplos em (37) evidenciam que o FST-MORPH hipergera. De fato, para cada verbo, o analisador constrói todos os três participios, independentemente de suas propriedades valenciais. Por exemplo, para verbos não transitivos, como *danser*, *arriver* e *ordonner*, são gerados participios passivos.<sup>12</sup> Por outro lado, formas como *dansé* e *arrivé* são caracterizadas pelo analisador como ambíguas entre PTPST+UNERG e PTPST+UNACC, quando só a primeira e a última são, respectivamente, válidas.

Qual a razão para essa hipergeração e quais as suas consequências no *parsing* sintático por meio da FrGramm 1.0? A hipergeração decorre de uma decisão nossa a respeito do *design* da gramática. Naturalmente teria sido possível, no âmbito da implementação do FST-MORPH, restringir a geração dos três tipos de participio com base nas duas propriedades sintáticas em jogo, a valência verbal (regência de um OBJ) e a seleção do auxiliar. De fato, a morfologia de estados finitos fornece meios de expressar essas restrições de forma elegante.<sup>13</sup> Dada a arquitetura da gramática da **Figura 3**, porém, a codificação de classes de valência na morfologia levaria a uma redundância na gramática, uma vez que, em LFG/XLE, essa informação precisa ser codificada nas formas semânticas dos verbos, nas respectivas entradas do componente lexical, como vimos em (8). Por outro lado, o fato de a morfologia hipergerar não implica necessariamente que a sintaxe hipergere. Isso pode ser evitado por meio de restrições na sintaxe que funcionem como filtro para as formas agramaticais da morfologia. Veremos mais adiante que a FrGramm implementa essas restrições, evitando a geração das construções do conjunto-teste negativo do tipo das exemplificadas em (23)-(32).

O módulo XLE-LEXICON possui três tipos de entradas lexicais. O primeiro tipo são entradas de formas plenas (*full-form entries*), que codificam as propriedades morfossintáticas de itens não contemplados na morfologia. Como vimos, na FrGramm 1.0, apenas verbos da 1ª conjugação estão codificados no transdutor lexical. Desse modo, os demais itens estão codificados como entradas de formas plenas. Em (38), temos a entrada para a forma *est*, 3ª pessoa do singular do indicativo do verbo pleno e do auxiliar *être*.

<sup>12</sup> Verbos geralmente possuem mais de uma valência. Não apassiváveis são as variantes dos verbos referidos sem um OBJ em sua moldura de subcategorização.

<sup>13</sup> As duas referidas restrições podem ser codificadas na gramática LEXC, por exemplo, por meio de *flag diacritics* (BEESLEY; KARTTUNEN, 2003).

(38)

- i. est V \* (^ PRED)=ÊTRE<(^ XCOMP)>(^ SUBJ)'
- ii. (^ SUBJ)=(^ XCOMP SUBJ) @(CAT (^ XCOMP) {AP PP})
- iii. (^ TENSE)=PRES @IND @(V-AGR 3 SG);
- iv. Aux \* { (^ CHECK PASS) =+\_ (^ PASSIVE) =c + (^ TENSE)=PRES |
- v. (^ VFORM) =c PART\_PAST (^ AUX) =c ÊTRE (^ CHECK ETRE) =+\_
- vi. (^ UNACC) =+ (^ TENSE)=COMPOUND\_PAST }
- vii. @(V-AGR 3 SG) @IND.

Em LFG/XLE, entradas lexicais para itens homônimos como *être* obedecem ao esquema geral de (39):

(39) *FORMA CATEGORIA1 SEPARADOR (ESQUEMAS FUNCIONAIS);  
CATEGORIA2 SEPARADOR (ESQUEMAS FUNCIONAIS).*

Nesse esquema, as expressões em caixa alta e itálico indicam os diferentes tipos de elementos constitutivos de uma entrada lexical. No caso de (38), a forma é *est*, as categorias são V (verbo) e Aux (auxiliar) e o separador é “\*”. Esses três elementos são obrigatórios. Os esquemas funcionais estão em (39) entre parênteses, para indicar que são facultativos.

Em (38), estão contemplados três usos de *être*. Em (i)-(iii), temos a variante que funciona como cópula na construção predicativa adjetival, que analisamos como verbo de alçamento. A linha (i) especifica a valência, como valor do atributo PRED. Trata-se de verbo que exige um SUBJ e um XCOMP. Observe que o SUBJ se encontra em (38) fora dos parênteses angulares. Isso indica que se trata de função gramatical subcategorizada pelo verbo, mas que não realiza argumento semântico do predicado verbal. Como vimos, a função XCOMP representa uma classe de complementos verbais com uma posição argumental aberta, a ser preenchida via controle por outra função gramatical do mesmo predicado. A linha (ii) caracteriza, inicialmente, essa variante como verbo de controle do sujeito. Em seguida, o predicado CAT do XLE determina que o XCOMP seja realizado como AP ou PP.<sup>14</sup> A linha (iii) especifica os traços flexionais: o tempo verbal, o modo e a concordância. Nessa mesma linha, temos a invocação de dois moldes, definidos no módulo XLE-TEMPLATES (**Figura 3**). O primeiro é o molde IND, que estabelece o modo indicativo. Em seguida, a invocação do molde V-AGR com os argumentos 3 e SG estabelece a concordância de 3ª pessoa do singular.

Nas linhas (iv)-(vi), temos, numa disjunção lógica delimitada por chaves, a segunda e terceira variantes, ou seja, os auxiliares da voz passiva (linha (iv)) e do passado composto (linhas (v)-(vi)). A invocação dos moldes V-AGR e IND na linha (vii) encontra-se fora da disjunção porque essas propriedades são comuns aos dois

<sup>14</sup> Por falta de espaço, não podemos explicar aqui todos os detalhes da notação do XLE. Para tanto, remetemos a Crouch et al. (2011).

auxiliares. Como categorias funcionais, ambos não possuem um atributo PRED, que, como vimos, codifica a informação semântica das categorias lexicais. Dessa forma, os auxiliares contribuem apenas com traços morfossintáticos para a estrutura F da sentença. As equações com o atributo CHECK em (iv) e (v) evitam a hipergeração em exemplos do tipo de (27), em que ocorre repetição agramatical de um auxiliar. Essa repetição é licenciada pelo caráter recursivo das regras de estruturação do VP (v. *infra*). O atributo CHECK foi proposto por King (2004) apenas para assegurar a boa formação sintática; em nada contribui para a descrição das propriedades gramaticais de uma sentença. A equação com o atributo PASSIVE restringe o emprego desse auxiliar a estruturas passivas. A última equação do auxiliar da passiva especifica que o tempo verbal é o presente.

O auxiliar do passado composto exige o particípio passado de verbo que seleciona o auxiliar *être* (linha (v)). A equação (^ UNACC) = + em (vi) força a concordância do particípio com o sujeito sentencial, em exemplos do tipo (1). A segunda equação especifica que o tempo verbal é o passado composto (SCHWARZE, 2001, p. 5).

Analogamente a (38), (40) codifica, para a 3ª pessoa do singular do presente do indicativo de *avoir*, tanto o uso quanto verbo pleno quanto como auxiliar. Diferentemente do auxiliar *être* do passado composto, especificado com o traço (^ UNACC) = + em (38), o auxiliar *avoir* é inespecificado para o atributo UNACC. A razão para isso é que o particípio, nesse caso, pode apresentar concordância ou não, dependendo do tipo de estrutura (cf. (4) e (6)).

- (40) a        V \* (^ PRED)=’AVOIR<(^ SUBJ)(^ OBJ)>’  
               (^ TENSE)=PRES    @(V-AGR 3 SG) @IND;  
       Aux \* (^ AUX) =c AVOIR    (^ VFORM) =c PART\_PAST  
               (^ TENSE)=COMPOUND\_PAST  
               (^ CHECK AVOIR) = +\_        @(V-AGR 3 SG) @IND.

O segundo tipo de entradas do módulo XLE-LEXICON tem como objeto as etiquetas da morfologia. As análises geradas pelo transdutor lexical não são diretamente interpretáveis pelo XLE. É necessário traduzir essas representações para esquemas funcionais. Em (41), reproduzimos as entradas desse tipo que se referem diretamente à implementação da passiva e do passado composto. As duas primeiras entradas invocam os moldes PAST e PASS, que são definidos em (42). A definição do molde PAST, por sua vez, invoca o molde ACT, definido em (43). O léxico inclui entradas para todas as etiquetas produzidas pelo transdutor, permitindo, por exemplo, que +F e +SG sejam convertidas nos traços GEN=FEM e NUM=SG.

- (41) +PTPST            V\_SFX XLE @PPAST.  
 +PTPASS            V\_SFX XLE @PASS.  
 +UNACC            V\_SFX XLE (^ UNACC) =c +.  
 +UNERG            V\_SFX XLE (^ UNACC) = -.
- (42) PASS = (^ PASSIVE) =c +.  
 PPAST = (^ VFORM)=PART\_PAST @ACT.
- (43) ACT = (^ PASSIVE) = -

Nas entradas do tipo de (41), o separador não é o asterisco “\*”, reservado a formas plenas como em (38), mas a palavra-chave “XLE”. As etiquetas +PTPST, +PTPASS etc., geradas pelo analisador morfológico, são tratadas como sufixos verbais pelo XLE, razão de a categoria desses elementos em (41) ser V\_SFX (do inglês *verbal suffix*). As equações funcionais atribuídas a esses sufixos são herdadas pelos verbos que os incorporam. Com isso, o particípio passivo exige um valor positivo para o atributo PASSIVE, ao passo que o particípio passado é especificado como VFORM=PART\_PAST e PASSIVE=-. Os dois tipos de particípio passado, por sua vez, são diferenciados por meio do valor do atributo UNACC. Se exigido um valor positivo (3ª linha de (41)), a concordância deve realizar-se; se definido um valor negativo (4ª linha de (41)), a concordância é bloqueada. As informações atribuídas desse modo aos três tipos de particípio, em interação com as entradas dos auxiliares (ver (38) e (40)), a regra lexical da passiva e as regras de estruturação sintagmática anotadas permitem que a gramática analise corretamente os exemplos do conjunto-teste positivo e reconheça como agramaticais os exemplos do conjunto-teste negativo do tipo de (23)-(32).<sup>15</sup>

O terceiro tipo de entrada do componente XLE-LEXICON é exemplificado pelas entradas de (44)-(46), que subjazem às variantes verbais de (1), (2) e (18), respectivamente.

- (44) arriver        V XLE @(UNACC\_V ARRIVER).  
 (45) danser        V XLE @(UNERG\_V DANSER).  
 (46) ordonner    V XLE @(DIRECTIVE ORDONNER OBJ2 DE).

Essas entradas codificam propriedades dos lemas das formas flexionadas geradas no componente morfológico, propriedades essas não contempladas pelas etiquetas morfológicas. No caso da FrGramm, essas propriedades adicionais são a forma lexical do verbo (que inclui a valência), a seleção do auxiliar e a concordância do particípio passado, entre outras. A invocação de moldes como os do **Quadro 2** permite especificar essas informações de forma bastante compacta. Cada um desses moldes codifica as propriedades comuns a todos os membros da classe. Por outro lado, as propriedades específicas de um membro particular da classe são especificadas por meio de parâmetros.

<sup>15</sup> Como veremos mais adiante, a atual versão da FrGramm não modela a precedência linear do auxiliar do passado composto relativamente ao auxiliar da passiva, analisando como gramaticais exemplos do tipo de (33) em que a ordem desses auxiliares está invertida.

Por exemplo, em (44) e (45), os moldes UNACC\_V e UNERG\_V são invocados com apenas um argumento, que é o lema do verbo. Em (46), contudo, o molde DIRECTIVE é invocado com três argumentos: o lema (ORDONNER), a função sintática controladora (OBJ2) e a forma do complementador (DE).

**Quadro 2** – Exemplos de moldes de classes valenciais na FrGramm 1.0

Molde	Parâmetros	Classe valencial	Auxiliar do passado composto	Concordância do particípio
UNACC_V	lema	verbos intransitivos inacusativos	ÊTRE	+
UNERG_V	lema	verbos intransitivos inergativos <sup>16</sup>	AVOIR	-
TRANS	lema	verbos transitivos	AVOIR	
DIRECTIVE	lema, controlador, forma do complementador	verbos diretivos	AVOIR	

Fonte: Elaboração própria.

A passivização é uma importante propriedade lexical. Como a FrGramm especifica que verbos são passivizáveis? As entradas em (47) respondem a essa pergunta.

- (47) *pele* V XLE @(PASSIVE @(TRANS PELE)).  
*forcer* V XLE @(PASSIVE @(DIRECTIVE FORCER OBJ À)).

Seguindo a implementação padrão da passivização em LFG/XLE (KING, 2004), essas entradas invocam o molde PASSIVE, que possui um único argumento: a invocação de um molde de classe valencial (**Quadro 2**). Trata-se, portanto, da aplicação de uma operação sobre o *output* de outra. Exemplifiquemos. A aplicação do molde TRANS sobre seu argumento gera os esquemas funcionais próprios dos verbos transitivos. Aplicado sobre esses esquemas, o molde PASSIVE realiza as transformações de (13), gerando, em interação com as informações codificadas nas entradas de sufixos verbais (ver (41)), entradas lexicais ativas e passivas.

Em (48), temos a definição do molde PASSIVE da FrGramm, uma adaptação do molde análogo proposto por King (2004) em sua gramática LFG/XLE do inglês.

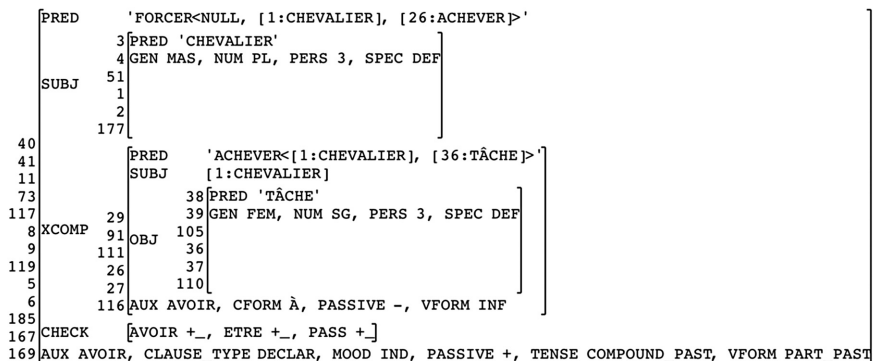
<sup>16</sup> No contexto da FrGramm, as designações *inacusativo* e *inergativo* têm caráter meramente mnemônico, uma vez que se referem não à semântica verbal, mas à seleção do auxiliar do passado composto. Ver nota 11.

(48) PASSIVE(\_SCHEMATA) =  
 { \_SCHEMATA (^ PASSIVE) = - |  
 \_SCHEMATA (^ PASSIVE) = + (^ OBJ) --> (^ SUBJ)  
 { (^ SUBJ) --> NULL | (^ SUBJ) --> (^ OBL) (^ OBL CASE) =c PAR } }.

Esse molde tem como único parâmetro um conjunto de esquemas funcionais (variável \_SCHEMATA). A definição do molde compreende uma disjunção: a primeira alternativa se refere à diátese ativa, a segunda, à diátese passiva. Essa última, por sua vez, compreende outra disjunção entre duas alternativas, dependendo da transformação do SUBJ em NULL ou em OBL. Nessa última alternativa, exige-se que o valor do atributo CASE (caso) do OBL seja PAR. Comum às duas variantes da passiva é a transformação do OBJ em SUBJ, que, no caso de verbos de controle do OBJ, como *forcer* em (5), ocorre tanto na forma lexical do verbo quanto na equação de controle funcional.

**Figura 5** – Estrutura F de (5) gerada pelo XLE a partir da FrGramm 1.0

"Les chevaliers ont été forcés à achever la tâche."



Fonte: Elaboração própria.

A esse respeito, comparem-se as estruturas F da **Figura 2**, **Figura 5** e **Figura 6**. Tanto na estrutura F da sentença ativa quanto nas estruturas F das respectivas passivas, o *parser* insere a estrutura F de *les chevaliers* como segundo argumento de FORCER, na predicação principal, e primeiro argumento de ACHEVER, na predicação secundária. Por outro lado, a estrutura F de *la reine* é inserida como primeiro lugar argumental de FORCER independentemente da realização desse argumento como SUBJ na **Figura 2** ou como OBL na **Figura 6**.

Como vimos, na arquitetura da LFG, a estrutura F de uma sentença constitui o *input* para a projeção da representação do significado sentencial. As análises da **Figura 2**, **Figura 5** e **Figura 6** evidenciam que a FrGramm produz as estruturas F esperadas para sentenças passivas e contrapartes ativas, que, no caso de verbos de controle do OBJ do tipo de *forcer*, constituem um desafio maior ao tratamento computacional do

que construções mais simples como (3) e (11). Essas estruturas F permitem, a partir da conversão dos valores do atributo PRED de cada sentença em formas lógicas, calcular as relações semânticas de acarretamento entre (4) e (5) e de equivalência entre (4) e (21) (CRUSE, 2000, p. 28-30).

**Figura 6** – Estrutura F simplificada de (21) gerada pelo XLE a partir da FrGramm 1.0

"Les chevaliers ont été forcés par la reine à achever la tâche

PRED	'	FORCER<	[ 26:REINE],	[ 1:CHEVALIER],	[ 32:ACHEVER >'
SUBJ	1	[	PRED	'	CHEVALIER']
OBL	26	[	PRED	'	REINE']
XCOMP	5	[	PRED	'	ACHEVER<
			SUBJ	[	1:CHEVALIER]
			OBJ	42	[
			PRED	'	TÂCHE']
			32	[	OBJ 42
			PRED	'	TÂCHE']

**Fonte:** Elaboração própria.

Retomemos agora a questão da hipergeração morfológica. Na sintaxe, as restrições propostas atuam como filtro das formas hipergeradas. Por exemplo, a análise morfológica de *arrivé* como particípio passado não flexionado (i.e. *arriver*+V+PTPST+UNERG) é bloqueada na sintaxe porque, por um lado, o sufixo +UNERG projeta UNACC=- (ver (41)); por outro lado, a entrada lexical de *arriver* em (44) projeta UNACC = +. Ora, essas duas especificações são incompatíveis, porque os valores do atributo UNACC, por serem diferentes, não unificam.

Em (48), em vez da equação (^ PASSIVE) = c +, que exige um valor positivo para o atributo PASSIVE, como propõe King (2004), temos a equação (^ PASSIVE) = +, que define esse valor como positivo, satisfazendo a exigência imposta pelos particípios passivos gerados na morfologia (ver (42)). Expliquemos, por meio da construção agramatical (32), como o molde da passiva filtra particípios passivos de verbos não transitivos, como *ordonner* na variante de (46). Para a forma *ordonnée*, o analisador morfológico gera as representações de (49) e (50). Pela entrada em (46), esse verbo, por meio do molde DIRECTIVE (**Quadro 2**), tem o atributo AUX especificado como AVOIR, o que exclui a primeira representação, porque a etiqueta +UNACC exige auxiliar com o traço UNACC=+ (ver (41)). Conforme (38), o auxiliar do passado composto é a única variante de *être* com essa especificação, porém, exige verbo com AUX=ÊTRE. A análise (50), por sua vez, é excluída porque +PTPASS exige PASSIVE=+ (ver (41)). No entanto, a única maneira de um verbo receber esse traço é por meio do molde da passiva em (48), o qual, conforme (46), não é aplicado sobre esse verbo.

(49) ordonner+V+PTPST+UNACC+F+SG

(50) ordonner+V+PTPASS+F+SG



O último módulo da arquitetura da **Figura 3** é XLE-SYN-RULES. Constitui-se de regras de estruturação sintagmática anotadas. Limitamo-nos aqui ao sintagma verbal. Seguindo Butt et al. (1999) e King (2004), mas divergindo de Schwarze (1998) e de Schwarze e Alencar (2016), distinguimos, com base no tipo de núcleo, entre VPv e VPaux, conforme (51). Nessa definição, VP é uma metacategoria, recurso do XLE que permite tanto expressar generalizações linguísticas quanto simplificar as regras de estrutura sintagmática e as estruturas C, uma vez que esse tipo de categoria não projeta nó na estrutura C. A **Figura 1** exemplifica os dois tipos de VP. Enquanto o primeiro é nucleado apenas por V, conforme (53), o segundo, definido em (52), é conucleado por Aux e V.<sup>17</sup> A motivação dessa distinção é excluir exemplos como (26), em que o verbo principal erroneamente precede o auxiliar. No entanto, noutras regras, como na própria regra (52), VPv e VPaux são intercambiáveis, fato que é capturado por meio da metacategoria VP.

(51)  $VP = \{ VPv \mid VPaux \}$ .

(52)  $VPaux \rightarrow Aux VP$ .

O exemplo (53) transcreve parte da regra do VPv, “[...]” indica os trechos suprimidos. Esse sintagma verbal se expande obrigatoriamente em um V e, facultativamente, em uma disjunção cujos membros representam as diferentes possibilidades de complementação verbal (não arroladas exaustivamente aqui), codificadas por meio das metacategorias IO, DO, OBL-PP e IC, definidas em (54).

(53)  $VPv \rightarrow V \{ DO \ IO \mid DO \mid IO \mid IO \ OBL-PP \dots \mid OBL-PP \mid IC \ DO \mid IC \dots \} \#0\#1 \dots$ .

(54)  $IO = PP: (\wedge OBJ2)=! \ (! \ CASE)=c \grave{A}$ .

$DO = NP: (\wedge OBJ)=!$ .

$OBL-PP = PP: (\wedge OBL)=!$ .

$IC = VP: (\wedge XCOMP)=!$ .

As definições de (54) consistem de categorias sintagmáticas anotadas com esquemas funcionais que especificam o tipo de função gramatical de cada categoria, a saber, OBJ2, OBJ, OBL e XCOMP, respectivamente. No caso da metacategoria IO, exige-se que o atributo CASE possua o valor  $\grave{A}$ . A metacategoria VP ocorre também do lado direito de (53), para dar conta dos complementos infinitivais de verbos de controle. Dado o caráter recursivo dessa expansão, construções bastante complexas com vários complementos encaixados e com mais de um auxiliar, como (17) ou (20), podem ser analisadas pela gramática.

Concluimos esta seção com a avaliação da FrGramm. Aplicado o respectivo *parser* gerado no XLE sobre o conjunto-teste positivo, todas as sentenças gramaticais rece-

<sup>17</sup> Sobre a noção de conúcleo (*co-head*) na LFG, ver Falk (2001, p. 39). Na teoria de Bresnan (2001, p. 132), conúcleos de natureza funcional são **núcleos estendidos** de uma categoria lexical.

beram as estruturas C e F esperadas. Apenas 8 sentenças receberam duas análises, devido à ambiguidade de anexação de um PP locativo, exemplificada em (34). A aplicação do *parser* ao conjunto-teste negativo, porém, revelou a necessidade de ajustes, na próxima versão da gramática, no módulo XLE-SYN-RULES. De fato, duas das 279 sentenças desse conjunto foram classificadas como gramaticais pelo *parser*. Trata-se de (33) e um outro exemplo estruturalmente análogo, em que a ordem dos auxiliares *avoir* e *être* está invertida; compare-se (33) com a construção gramatical em (16). Isso mostra que a FrGramm 1.0 hipergera nesse aspecto, não modelando a relação de precedência entre esses dois auxiliares, uma vez que *avoir* deve preceder *être* quando ambos funcionam como auxiliares de um dado verbo principal.

### Considerações finais

Neste artigo, descrevemos o tratamento da passiva e do passado composto na FrGramm 1.0, uma gramática computacional do francês de média cobertura sintática que recentemente implementamos em LFG/XLE. Devido à duplicidade de auxiliares e à concordância do particípio no passado composto, esse fenômeno gramatical apresenta maior complexidade em francês do que em línguas como português e inglês. Por outro lado, a análise dessas construções bem como da construção predicativa adjetival, superficialmente análoga à passiva, tem sido objeto de controvérsias na teoria da LFG.

A FrGramm é a única gramática do francês desse porte implementada em LFG que está acessível *on-line* de forma irrestrita, sob licença que permite a redistribuição de modificações. Desse modo, constitui plataforma para testagem das propriedades computacionais das diferentes abordagens teóricas dessas construções no formalismo LFG/XLE, podendo também ser adaptada para outros sistemas.

Frank (1996) e Schwarze (1998) são os dois fragmentos de gramática do francês anteriores diretamente comparáveis à nossa abordagem, por estarem documentados de forma suficientemente detalhada para permitir uma reimplementação no XLE. Como a FrGramm se posiciona em relação a essas duas propostas? Em primeiro lugar, a FrGramm tem cobertura muito mais ampla do que o fragmento de Schwarze (1998), que não inclui o passado composto. Esse trabalho não explicita se a regra da passiva, tanto em sua dimensão morfológica quanto lexical, foi de fato implementada.

O fragmento de Frank (1996) é bem mais abrangente que o recorte gramatical modelado na FrGramm. Enquanto nosso fragmento se restringe a sentenças declarativas com os constituintes em sua ordem canônica, o de Frank (1996) inclui sentenças interrogativas, relativas e várias outras construções com deslocamento de constituintes. Isso permite à gramática de Frank (1996) contemplar a concordância do PTPST com o OBJ. Seguindo a técnica de desenvolvimento em espiral, esse fenômeno teve de ficar fora da primeira versão da FrGramm, dada a complexidade do tratamento dessas construções.

Uma deficiência importante da gramática de Frank (1996) é não integrar um analisador morfológico, limitando-se a um léxico de formas plenas. Desse modo, cada particípio passivo é codificado individualmente no léxico, por meio de um molde específico para cada classe valencial. Consequentemente, não há nessa abordagem uma única regra da passiva.

A FrGramm preenche essas lacunas da abordagem de Frank (1996). Incorpora um transdutor lexical que analisa as formas de um grupo de verbos da 1ª conjugação, o que possibilita a integração entre morfologia e léxico na modelação computacional da regra da passiva. Uma única regra de passivização contempla todas as classes de valência passivizáveis. Várias classes de valência foram implementadas, incluindo cópulas e verbos de controle do objeto. Esses últimos representam uma dificuldade extra para o tratamento computacional, uma vez que o controlador, na passiva, passa a ser o sujeito. A FrGramm contempla os aspectos morfológicos, sintáticos e semânticos da passiva enquanto processo lexical produtivo, produzindo estruturas F adequadas tanto para construções simples quanto para estruturas de controle.

Dadas essas características, a FrGramm, abstraindo-se a menor cobertura sintática, revela-se superior à proposta de Frank (1996) em modularidade e escalabilidade. A FrGramm possui uma única regra da passiva, válida para todas as classes de valência apassiváveis, enquanto a abordagem de Frank (1996) pressupõe uma regra para cada classe. Essa diferença se revela decisiva na ampliação do léxico. Na FrGramm, para dar conta de sentenças com um verbo como *avancer* ‘avançar’, por exemplo, basta incluir a raiz *avanc* no componente ROOTS1 da morfologia e especificar as valências no léxico de lemas, nos moldes de (44)-(47). A inclusão de novos itens lexicais na gramática de Frank (1996) é bem mais trabalhosa.

A avaliação da FrGramm produziu resultados bastante satisfatórios. O respectivo *parser* analisou corretamente as 157 sentenças gramaticais do conjunto-teste positivo, que compreende exemplos na voz ativa e passiva tanto no presente do indicativo quanto no passado composto. As estruturas F geradas para todos os exemplos gramaticais foram as esperadas. Das 279 sentenças agramaticais do conjunto-teste negativo, apenas duas não foram analisadas corretamente. A gramática lhes atribui uma estrutura F válida. Essas duas sentenças estão no passado composto da voz passiva. A razão para essa deficiência é que a atual versão da FrGramm não modela a relação de precedência linear entre os auxiliares *avoir* e *être* nesse tipo de exemplo. Esse problema será sanado na próxima versão da gramática, que terá sua cobertura ampliada para dar conta da concordância do PTPST com o OBJ.

Com isso, esperamos contribuir para o debate, no âmbito da LFG, a respeito da estrutura predicacional das duas construções verbais perifrásticas, analisadas neste artigo como monopredicacionais. De fato, essa nova versão da gramática pode ser adaptada para implementar a análise bipredicacional, permitindo comparar, no XLE, a complexidade computacional das duas abordagens concorrentes.

## Agradecimentos

Agradecemos a Valeria de Paiva pela intermediação junto a John Maxwell e Daniel Bobrow, ambos do Palo Alto Research Center da Xerox (PARC), para obtenção de uma licença não comercial gratuita do XLE. Nossos agradecimentos se estendem a essa empresa bem como a todos os pesquisadores envolvidos na criação desse software. Somos também grato a Christoph Schwarze, Jessé de Sousa Mourão e os revisores anônimos pelos valiosos comentários e sugestões a respeito de versões anteriores deste artigo. No entanto, todos os erros remanescentes são de nossa responsabilidade.

ALENCAR, L. A computational implementation of periphrastic verb constructions in French. *Alfa*, São Paulo, v.61, n.2, p.351-380, 2017.

- *ABSTRACT: This paper describes the treatment of passive and compound past tense in FrGramm, a computational grammar of French, implemented within Lexical-Functional Grammar (LFG) using the XLE software. Due to the dual auxiliary system and past participle agreement, the latter periphrasis manifests greater structural complexity and therefore presents a greater challenge to computational implementation in French than in languages such as English and Portuguese. An additional difficulty is modeling the morphological and syntactico-semantic regularities of the passive construction. In FrGramm, this problem is solved by means of a productive lexical rule. FrGramm also implements the constraints governing the building of both verbal periphrases, excepting participle object agreement. The implementation was evaluated by applying a parser to a set of 157 grammatical sentences and a set of 279 ungrammatical sentences. All sentences from the former set were correctly parsed. Only two constructions from the latter set that violate the linear precedence of the compound past auxiliary over the passive auxiliary were analyzed as grammatical. FrGramm is the only LFG grammar of French with similar coverage that is freely available on-line. A future version will handle participle object agreement and also avoid the mentioned overgeneration.*
- *KEYWORDS: Computational linguistics. Deep syntactic parsing. Lexical-Functional grammar. LFG/XLE. Finite-state morphology. French verbal periphrases. Passive voice.*

## REFERÊNCIAS

BEESLEY, K. R.; KARTTUNEN, L. **Finite state morphology**. Stanford: CSLI, 2003.

BEST, J. IBM Watson: the inside story of how the Jeopardy-winning supercomputer was born, and what it wants to do next. **TechRepublic**, 2013. Disponível em: <<https://www.techrepublic.com/article/ibm-watson-the-inside-story-of-how-the-jeopardy-winning-supercomputer-was-born-and-what-it-wants-to-do-next/>>. Acesso em: 16 abr. 2016.

- BOULLIER, P.; SAGOT, B.; CLÉMENT, L. Un analyseur LFG efficace pour le français: SxLfg. In: TRAITEMENT AUTOMATIQUE DES LANGUES NATURELLES, 12., 2005. *Actes...* Dourdan, 2005. p.403-408. Disponível em: <[http://www.atala.org/taln\\_archives/TALN/TALN-2005/taln-2005-court-004](http://www.atala.org/taln_archives/TALN/TALN-2005/taln-2005-court-004)>. Acesso em: 9 fev. 2016.
- BRESNAN, J. **Lexical-functional syntax**. Malden: Blackwell, 2001.
- BUSSMANN, H. (Org.). **Lexikon der Sprachwissenschaft**. 3.ed. Stuttgart: Kröner, 2002.
- BUTT, M. et al. **A grammar writer's cookbook**. Stanford: CSLI, 1999.
- CLÉMENT, L. **XLFG**. Bordeaux: University Bordeaux, 2014. Disponível em: <<http://www.xlfg.org/>>. Acesso em: 22 fev. 2016.
- CLÉMENT, L.; KINYON, A. XLFG – an LFG Parsing Scheme for French. INTERNATIONAL LEXICAL-FUNCTIONAL GRAMMAR CONFERENCE, 6., 2001. **Proceedings...** Stanford: CSLI, 2001. p.47-65.
- CROUCH, D. et al. **XLE documentation**. Palo Alto: Palo Alto Research Center, 2011. Disponível em: <[http://www2.parc.com/isl/groups/nlft/xle/doc/xle\\_toc.html](http://www2.parc.com/isl/groups/nlft/xle/doc/xle_toc.html)> Acesso em: 22 fev. 2016.
- CRUSE, D. A. **Meaning in language: an introduction to semantics and pragmatics**. Oxford: Oxford University Press, 2000.
- FALK, Y. N. **Lexical-functional grammar: an introduction to parallel constraint-based syntax**. Stanford: CSLI, 2001.
- FRANCEZ, N.; WINTNER, S. **Unification grammars**. Cambridge: CUP, 2012.
- FRANK, A. Eine LFG-Grammatik des Französischen. In: BERMAN, J.; FRANK, A. **Deutsche und französische Syntax im Formalismus der LFG**. Tübingen: Niemeyer, 1996. p.97-244.
- KAPLAN, R. M.; BRESNAN, J. Lexical-functional grammar: a formal system for grammatical representation. In: BRESNAN, J. (Org.). **The mental representation of grammatical relations**. Cambridge: MIT Press, 1982. p.173-281.
- KING, T. H. **Starting a ParGram grammar**. 2004. Disponível em: <<http://www2.parc.com/isl/groups/nlft/xle/doc/PargramStarterGrammar/starternotes.html>>. Acesso em: 10 nov. 2012.
- MCCORD, M. C.; MURDOCK, J. W.; BOGURAEV, B. K. Deep parsing in Watson. **IBM Journal of Research and Development**, Armonk, v. 56, n. 3/4, p. 1-15, 2012.
- MÜLLER, S. **Grammatical theory: from transformational grammar to constraint-based approaches**. Berlin: Language Science Press, 2016. Disponível em: <<http://langsci-press.org/catalog/book/25>>. Acesso em: 22 mar. 2016.

PALMER, D. D. Text preprocessing. In: INDURKHYA, N.; DAMERAU, F. J. (Org.). **Handbook of natural language processing**. 2.ed. Boca Raton, Florida: Chapman & Hall/CRC, 2010. p.9-30.

PATEJUK, A.; PRZEPIÓRKOWSKI, A. In favour of the raising analysis of passivisation. INTERNATIONAL LEXICAL-FUNCTIONAL GRAMMAR CONFERENCE, 19., 2014. **Proceedings...** Stanford: CSLI, 2014. p.461-481.

PRATT-HARTMANN, I. Computational complexity in natural language. In: CLARK, A.; FOX, C.; LAPPIN, S. (Org.). **The handbook of computational linguistics and natural language processing**. Malden: Wiley & Blackwell, 2010. p.43-73.

SAGOT, B. **Page web de Benoît Sagot – équipe Alpage (INRIA/Paris 7): SxLFG**. Paris: Université Paris Diderot, [2015?]. Disponível em: <<http://alpage.inria.fr/~sagot/sxlf.html>>. Acesso em: 22 fev. 2016.

SCHWARZE, C. Do sentences have tense? In: INTERNATIONAL LEXICAL-FUNCTIONAL GRAMMAR CONFERENCE, 6., 2001. **Proceedings...** Stanford: CSLI, 2001. p.449-463.

SCHWARZE, C. **Lexikalisch-funktionale Grammatik: eine Einführung in 10 Lektionen mit französischen Beispielen**. 2.ed. Konstanz: Fachgruppe Sprachwissenschaft der Universität Konstanz, 1998.

SCHWARZE, C.; ALENCAR, L. F. de. **Lexikalisch-funktionale Grammatik: eine Einführung am Beispiel des Französischen mit computerlinguistischer Implementierung**. Tübingen: Stauffenburg, 2016.

SULGER, S. et al. ParGramBank: the ParGram parallel treebank. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS, 51., 2013. **Proceedings...** Sofia: Association for Computational Linguistics, 2013. p.550-560.

ZELLE, J. M. **Python programming: an introduction to computer science**. Wilsonville: Franklin, Beedle & Associates, 2004.

ZWEIGENBAUM, P. Un analyseur syntaxique pour grammaires lexicales-fonctionnelles. **T.A. Informations**, Paris, v. 32, n. 2, p.19-34, 1991. Disponível em: <<https://perso.limsi.fr/pz/FTPapiers/ZweigenbaumTAI91.pdf>>. Acesso em: 10 fev. 2016.

Recebido em abril de 2016

Aprovado em janeiro de 2017