FRIGRAM: a French Interaction Grammar

Guy Perrier

FRIGRAM: a French Interaction Grammar

Guy Perrier

ISSN 0249-6399   ISRN INRIA/RR--8323--FR+ENG

RESEARCH REPORT
N° 8323

December 2014

Project-Team Sémagramme

FRIGRAM: a French Interaction Grammar

Guy Perrier[*]

Project-Team Sémagramme

Research Report n° 8323 — December 2014 — 257 pages

Abstract: The report is a documentation for the French grammar FRIGRAM, which is grammar with a large coverage written in the formalism of Interaction Grammar. The originality of the formalism lies in its system of polarities, which expresses the resource sensitivity of natural languages and which is used to guide syntactic composition. The version of the grammar that is documented here is 3.0.0 and it is freely available at the URL `frig.loria.fr`.

Key-words: formal grammar, syntax, French grammar, polarity, tree description, Categorial Grammar, Interaction Grammar

[*] Thanks to Bruno Guillaume for his help in the construction of FRIGRAM and the review of this report

RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

FRIGRAM : une grammaire d'interaction du français

Résumé : Ce rapport est une documentation pour la grammaire du français FRIGRAM, qui est une grammaire à large couverture écrite dans le formalisme des grammaires d'interaction. L'originalité du formalisme réside dans son système de polarités qui exprime la sensibilité aux ressources des langues naturelles et qui est utilisé pour guider la composition syntaxique. La version de la grammaire qui est documentée ici est la 3.0.0 et elle est librement disponible à l'URL `frig.loria.fr`.

Mots-clés : grammaire formelle, syntaxe, grammaire du français, polarité, description d'arbre, grammaire catégorielle, grammaire d'interaction

Contents

Chapter 1

Generalities

FRIGRAM is written in the framework of the Interaction Grammar (IG) formalism. The originality of the formalism lies in its system of polarities, which expresses the resource sensitivity of natural languages and which is used to guide syntactic composition.

The objects manipulated by the formalism are *Polarized Tree Descriptions (PTDs)*. A PTD is an underspecified tree of constituents, where nodes are decorated with polarized features expressing the morpho-syntactic properties of constituents.

A polarized feature is a triple *(name, polarity, value)*. Polarities express the ability of a PTD to interact with other PTDs. Among all features, two play a particular role:

- `cat` gives the syntactic category of the constituent associated with the feature;
- `funct` gives the syntactic function of the constituent; if the value of the feature is `void`, it means that the constituent has no syntactic function in the sentence.

A grammar is defined as a finite set of PTDs called *Elementary Polarized Tree Descriptions (EPTDs)*.

For a complete presentation of the formalism, the reader can refer to [GP09].

1.1 The principles of the grammar

FRIGRAM includes about 4000 EPTDs, which all respect some principles. There are two kinds of principles: the principles verified by each EPTD individually and the principles verified by the models of PTDs representing the syntax of sentences.

1.1.1 The principles for individual descriptions

Definition 1 *A node with a positive or saturated* `cat` *feature is called a* concrete node*.*

Principle 1 (cat-funct) *In an EPTD, any node has a* `cat` *feature and if it is concrete, it has also a* `funct` *feature.*

The consequence is that any node of a model has a `cat` feature and a `funct` feature. Another consequence is that any node of a model has a unique concrete antecedent in the original PTD, because two concrete nodes of a PTD cannot merge in the model, according to the composition rules of polarities.

Principle 2 (strict lexicalisation) *Any EPTD has exactly one anchor node. This anchor node has a saturated* `cat` *feature with an atomic feature value.*

Definition 2 *A* spine *in an EPTD is a list of nodes $N_1, N_2, \dots, N_p$ such that:*

- *for any i such that $1 < i \leq p$, node N_i is a daughter node of N_{i-1};*
- *for any i such that $1 < i \leq p$, node N_i has a saturated feature* `cat` *and a feature* `funct` $\leftrightarrow$ `head`*;*
- *node N_1 is a concrete node and its feature* `funct` *has a value different from* `head`*; it is called the* maximal projection *of all nodes belonging to the spine;*
- *node N_p is either an anchor or an empty leaf; in the first case, the spine is called a* main spine*; in the second case, it is called an* empty spine*; in both cases, node N_p is called the* lexical head *of all nodes belonging to the spine.*

Principle 3 (spine) *Any concrete node of an EPTD belongs to exactly one spine.*

A corollary of the strict lexicalisation and spine principles is that every EPTD as exactly one main spine.

An important corollary of the spine principle is that every node N of a PTD model has exactly one lexical head in this model, denoted $head(N)$ and defined as follows: the concrete antecedent of N in the initial PTD belongs to exactly one spine and $head(N)$ is the interpretation in the model of the leaf ending the spine.

A second important corollary is that every node in a PTD model which is not a leaf has exactly one daughter node with the feature `funct : head`. By following all nodes with this feature, we have a more direct way of finding the lexical head of every node in a PTD model.

A third corollary is that each node of an EPTD with a positive feature `cat` is the maximal projection of some spine.

The spine definition and principle are illustrated with three EPTDs used to parse the following sentences.

(1.1) *Jean* *arrive* *plus tôt* **qu'** *hier* *.*
Jean is coming earlier than yesterday .
Jean is coming earlier than yesterday.

(1.2) **Qui** *dort* *dîne* *.*
Who sleeps has dinner .
Who sleeps has dinner.

(1.3) *Jean* *rencontre* *Marie* *dans* *l'entreprise* *de* **qui** *travaille* *Pierre* *.*
Jean is meeting Marie in the company of whom works Pierre .
Jean is meeting Marie in whose company Pierre works.

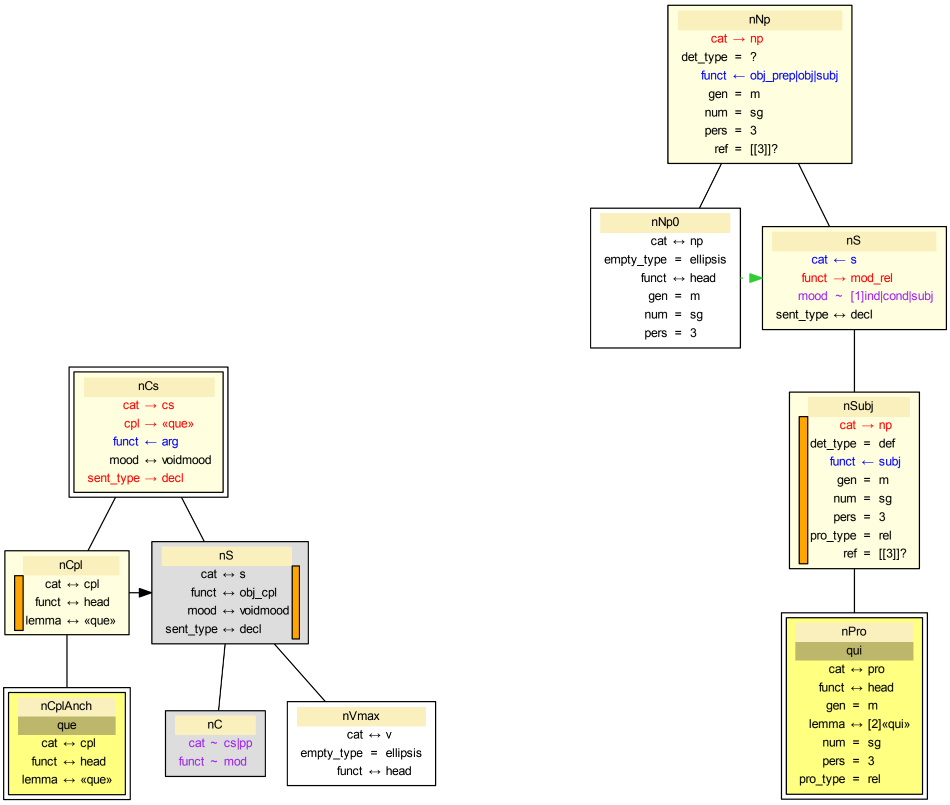

Figure 1.1: EPTD associated with the complementizer *que* introducing a comparison clause and EPTD associated with the subject relative pronoun *qui* used without antecedent

Figure 1.1 shows the EPTDs associated with the words in bold in the two first sentences. In the left EPTD, there is a main spine *nCs*, *nCpl*, *nCplAnch* and an empty spine *nS*, *nVmax*. Node *nCs* is the maximal projection of the main spine and *nCplAnch* its lexical head. In the same EPTD, node *nS* is the maximal projection of the empty spine and *nVmax* its lexical head. In the right EPTD, there is a main spine *nSubj*, *nPro* and an empty spine *nNp*, *nNp0*. Node *nSubj* is the maximal projection of the main spine and *nPro* its lexical head. In the same EPTD, node *nNp* is the maximal projection of the empty spine and *nNp0* its lexical head.

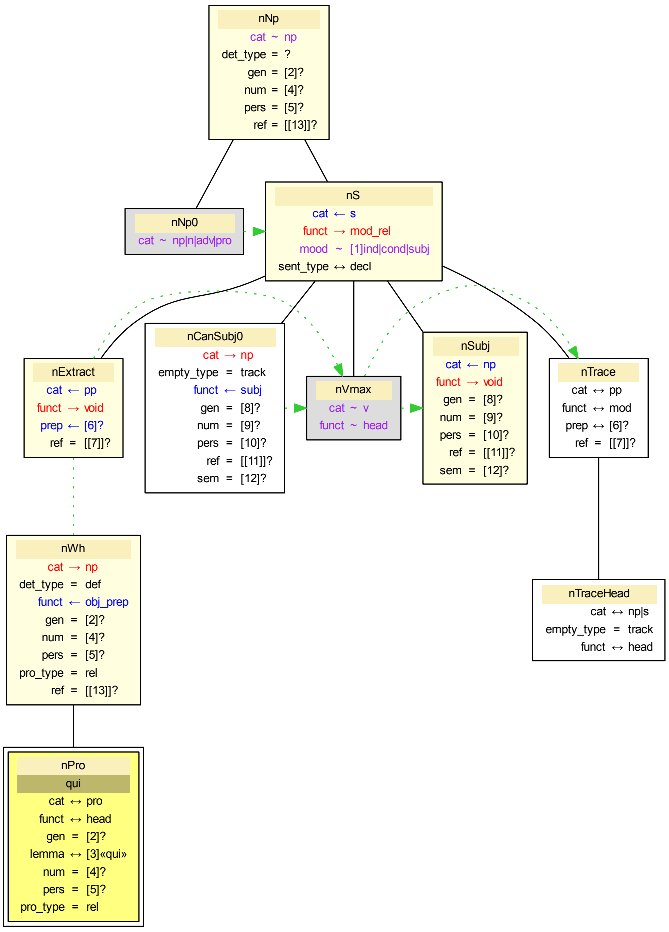

Figure 1.2: EPTD associated with the relative pronoun *qui* used in an extracted indirect complement

The EPTD of Figure 1.2 is associated with the relative pronoun *qui* used in an extracted indirect complement, as in Sentence (1.3). It includes three spines: the main spine *nWh*, *nPro* and two empty spines, one reduced to a single node (*nCanSubj0*) and the other one with two nodes *nTrace*, *nTraceHead*.

1.1.2 The principle for models

Principle 4 (function unicity) *For any node of a model which is not a leaf and for any* `funct` *feature different of* `mod` *and* `iobj`*, there is at most one daughter node with such a* `funct` *feature.*

This principle is useful for guiding the parsing with IG.

1.2 The organisation of the grammar

1.2.1 The interface with the lexicon

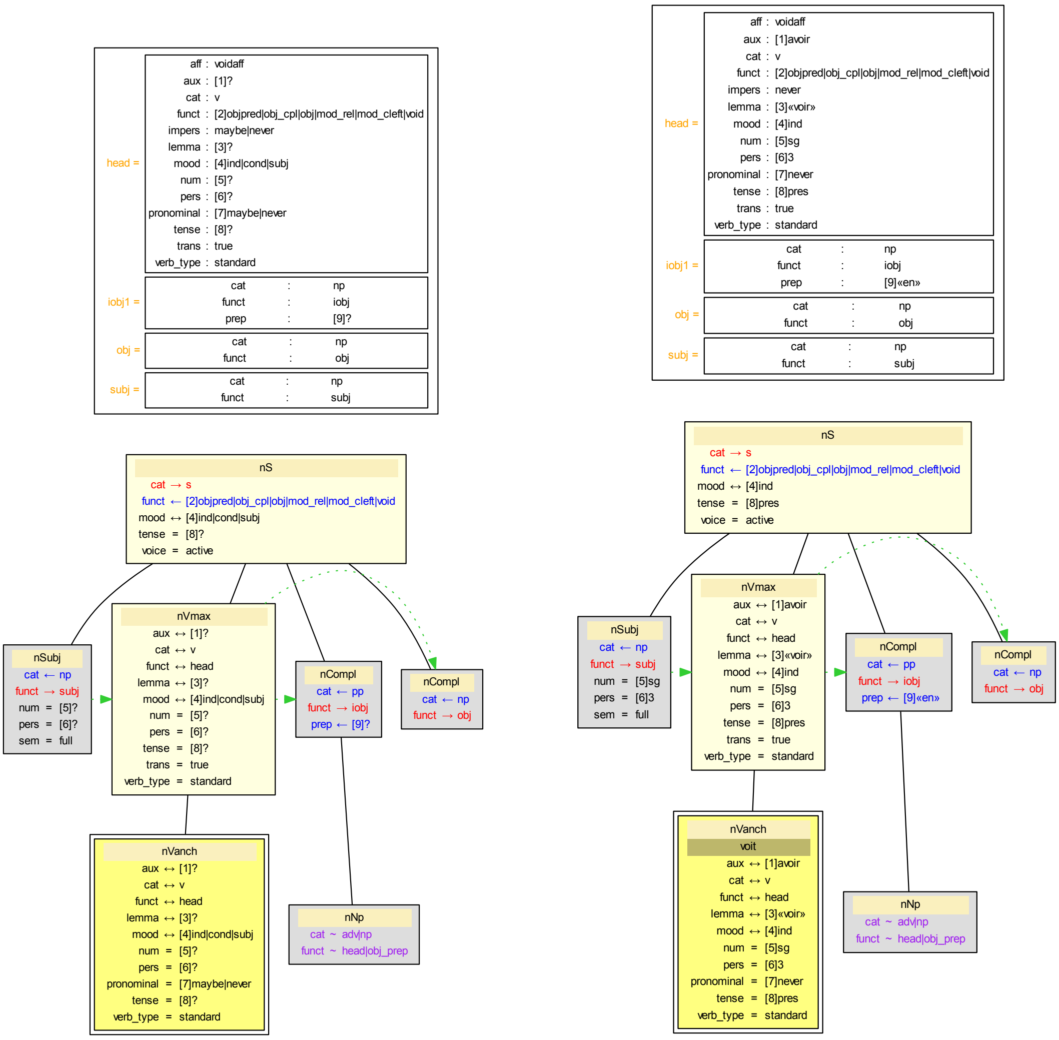

Figure 1.3: From left to right, a non anchored EPTD describing the syntactic behaviour of a transitive verb in the active voice, and the same EPTD after anchoring with the verb *voit*

FRIGRAM is strictly lexicalized: each EPTD of the grammar has a unique anchor node intended to be linked with a word of the language. For this, it is associated to a feature structure describing a syntactic frame corresponding to words able to anchor it, the description being independent of the grammatical formalism. This feature structure constitutes the *interface* of the EPTD with the lexicon.

The set of features used in the interfaces differs from the one used in EPTDs because they do not play the same role: they do not aim at describing syntactic structures but they are used for describing the morpho-syntactic properties of the words of the language in a way independent of the formalism.

On the left part of Figure 1.3, an EPTD represents the syntactic behavior of a transitive verb, which also requires an indirect complement, in the active voice and in a mood which is conditional, indicative or subjunctive. On the top, there is its interface, which expresses these properties with a two level feature structure:

- At the top level of the feature structure, the features `head`, `iobj1`, `obj` and `subj` indicate the different components of the frame required from verbs anchoring the EPTD. They mean that these verbs must have a subject, a direct object and an indirect object.
- The second level gives morpho-syntactic properties of each element of the top level[1].

The lexicon which is linked to the grammar must have the same form for its entries as for the EPTD interfaces because the EPTD anchoring is performed by feature filtering. For instance, the feature structure on the right part of Figure 1.3 represents a possible lexical entry for the verb *voit*. It succeeds to filter the interface on the left and a side effect is to instantiate feature values shared by the EPTD and its interface, the values of the `lemma`, `mood`, `num`, `pers`, `pronominal` and `tense` features. The result is the anchored EPTD on the right part of the figure.

1.2.2 The source grammar as a hierarchy of classes

The about 4000 EPTDs of FRIGRAM have not been written one by one but they are automatically generated from a source grammar, which is structured as a hierarchy of classes, built each one from the other ones with three operations: *simple inheritance*, *conjunction* and *disjunction*.

In the following, when there is ambiguity, we call this source grammar FRIGRAM$_S$ whereas the object grammar, constituted of the EPTDs, is called FRIGRAM$_O$. The compilation of FRIGRAM$_O$ from FRIGRAM$_S$ is performed by XMG [CDG$^+$13], which is a software dedicated to the design of electronic grammars usable in NLP.

Definition 3 *A* terminal class *of the source grammar is a class that is evaluated to produce the corresponding EPTDs of the object grammar.*

[1] For the explanation of the different feature names, see the first section of the chapter about verbs.

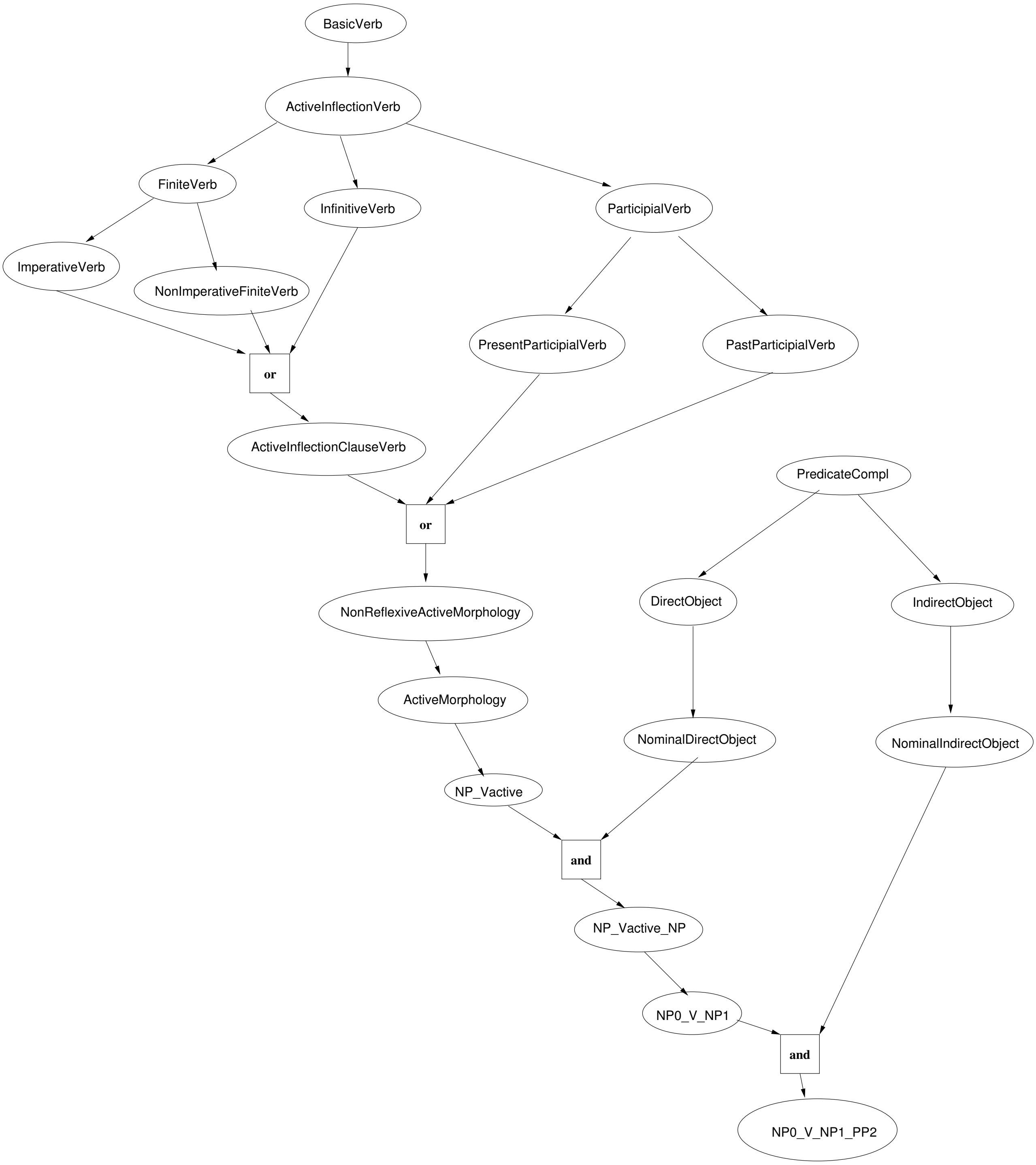

Figure 1.4: Partial view of the hierarchy of classes used for defining the terminal class NP0_V_NP1_PP2

Of course, all classes that are not operands for one of the three operations are terminal classes. Figure 1.4 gives a partial view of the hierarchy of the 40 classes used for producing the terminal class NP0_V_NP1_PP2. Classes related to impersonal and reflexive constructions as well as passive and middle voices are not considered in this view. The evaluation of the NP0_V_NP1_PP2 class produces 58 EPTDs.

FRIGRAM$_S$ includes about 400 classes and among them 160 terminal classes defining around 4000 EPTDs. It means that every terminal defines 25 EPTDs on average.

1.2.3 The grouping of classes by modules

The 400 classes of FRIGRAM$_S$ are grouped by modules. Here is the list of all modules in the alphabetic order:

- ADJECTIVE: classes concerning adjectives,
- ADVERB: classes concerning adverbs,
- COMPLEMENT: classes modelling all kinds of complements required by verbs, nouns or adjectives
- COMPLEMENTIZER: classes concerning complementizers[2],
- COORDINATION: classes modelling coordination,
- DETERMINER: classes concerning determiners, except interrogative determiners,
- EXTRACTGRAMWORD: classes related to the phenomenon of extraction (from relative, interrogative and cleft clauses)
- INTERROGATIVE: classes concerning interrogative pronouns, adverbs and determiners,
- NOUN: classes concerning common and proper nouns,
- PREPOSITION: classes concerning prepositions,
- PROCLITIC: classes concerning clitic pronouns,
- PRONOUN: classes concerning lexical pronouns, except interrogative and relative pronouns,
- PUNCTUATION: classes concerning punctuation signs,
- RELATIVE: classes concerning relative pronouns,
- VERB: classes defining the different families of verbs according to their subcategorization frame and specific verbs as presentatives and modal and causative verbs,

[2] The prepositions *à* and *de* introducing direct object infinitives are considered as complementizers.

- VERBKERNEL: classes defining the common verbal kernel of all verbs with the morphology and its interaction with the form of the subject, the syntactic function of the verb and its voice,
- VERBIMPERSONALDIATHESES: classes modelling the different diatheses, active, passive and middle, with an impersonal subject,
- VERBPERSONALDIATHESES: classes modelling the different diatheses, active, passive and middle, with a personal subject,
- VERBSUBJECTCONTROL: classes modelling the control of infinitive subjects by arguments of the verb governing the infinitive.

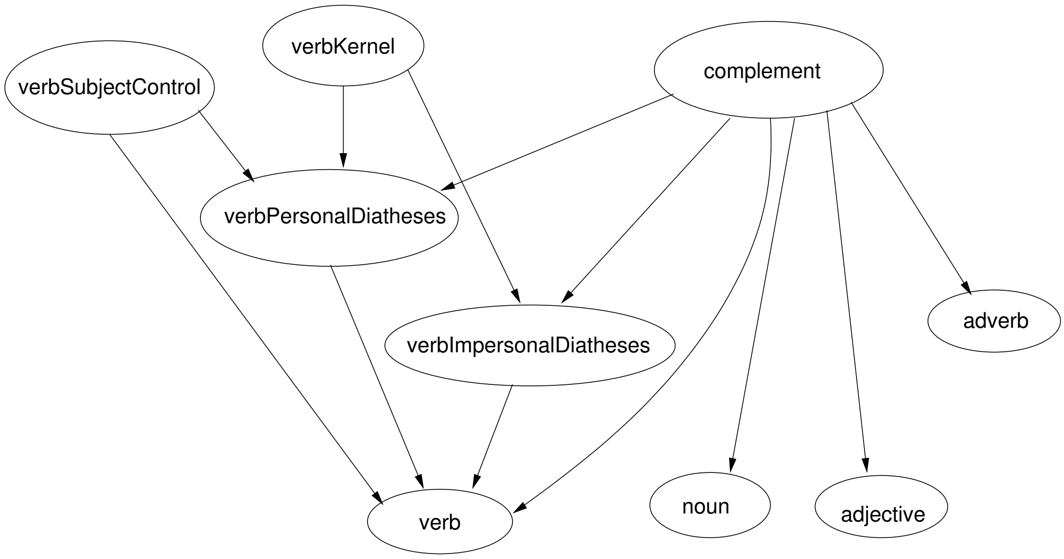

Figure 1.5: The hierarchy of modules grouping the classes of FRIGRAM$_S$ concerning verbs, nouns, adjectives and adverbs

Some classes of one module are defined from classes of another module. We can represent it with a graph where an edge means that some classes of the target module are defined from classes of the source module. Figure 1.5 shows these dependencies for the modules concerning verbs, nouns and adjectives.

Figure 1.6 shows these dependencies for the modules modelling extraction from relative, interrogative and cleft clauses. The modules absent from Figure 1.5 and Figure 1.6 are isolated modules without external dependencies.

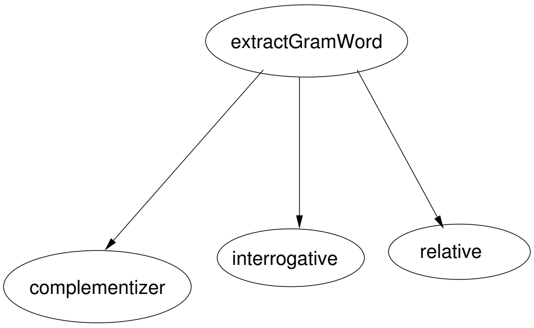

Figure 1.6: The hierarchy of modules grouping the classes of FRIGRAM$_S$ concerning extraction

Chapter 2

Complements

The Complement module gathers classes describing complements required by verbs, nouns and adjectives. That is why the following modules depend on the Complement module, as Figure 1.5 in the previous chapter shows it: verbPersonalDiatheses, verbImpersonalDiatheses, noun and adjective. A basic class, PredComplement, describes the common features of all complement classes. It generates the PTD[1] of Figure 2.1.

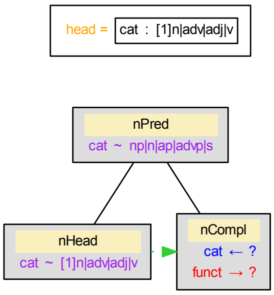

Figure 2.1: The PTD defined by the PredComplement class

The negative `cat` feature and the positive `funct` feature express that the complement represented with node *nCompl* is required by its head represented with node *nHead*.

Complement are of three kinds: direct objects, predicate complements and indirect objects.

[1] It is not an EPTD (no anchor). It has to be extented by some other classes to produce EPTDs.

2.1 Direct objects

Only verbs have direct objects. Here are various examples of direct objects[2].

(2.1) *Jean interroge Marie .*
Jean is asking Marie .
Jean is asking Marie.

(2.2) *Ce colis pèse lourd .*
This parcel weights a lot .
This parcel weights a lot

(2.3) *Jean veut venir .*
Jean wants to come .
Jean wants to come.

(2.4) *Jean apprend à travailler .*
Jean learns working .
Jean learns working.

(2.5) *Jean veut que Marie vienne .*
Jean wants that Marie comes .
Jean wants that Marie comes.

(2.6) *Jean demande quand Marie vient .*
Jean asks when Marie is coming .
Jean asks when Marie is coming.

The different forms of direct objects correspond to different classes:

- NominalObject for nominal objects (Sentence (2.1)),
- AdverbialObject for adverbial objects (Sentence (2.2)),
- DirectInfinitiveClauseObject for direct object infinitives (Sentence (2.3)),
- IndirectInfinitiveClauseObject for object infinitives introduced with a preposition used as a complementizer (Sentence (2.4)),
- DeclarativeFiniteClauseObject for declarative object finite clauses (Sentence (2.5)),
- InterrogativeClauseObject for interrogative object finite clauses (Sentence (2.6)).

[2] For Example (2.2), since the complement has a very particular behavior, some linguists exclude it from direct objects.

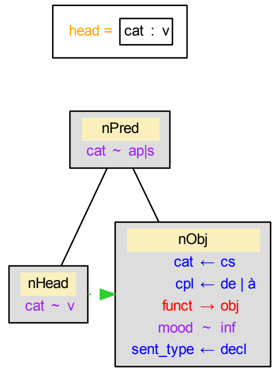

Figure 2.2: PTD defined by the IndirectInfinitiveClauseObject class

There is a common ancestor of all these classes, the DirectObject class. The NominalObject and AdverbialObject classes directly inherit the DirectObject class. For the other classes, there is an intermediate class, the ClauseObject class. Since the IndirectInfinitiveClauseObject class and the DeclarativeFiniteClauseObject class often have a similar behavior, there is a class grouping the two cases, the DeclarativeComplementedClauseObject.

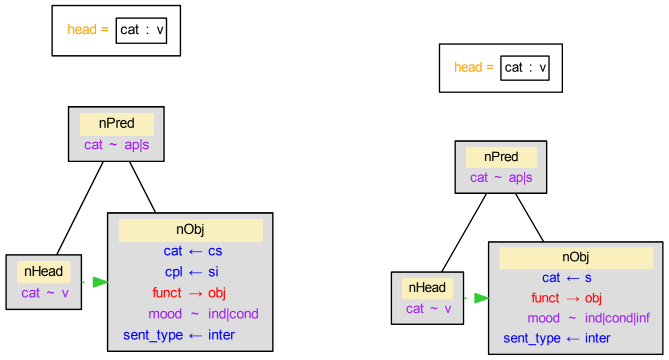

Figure 2.3: The two PTDs defined by the InterrogativeClauseObject class

Figure 2.2 shows the PTD defined by the IndirectInfinitiveClauseObject class. For this class, the object is an infinitive clause introduced with *à* or *de* considered as complementizers. This is expressed with the negative feature `cpl` ← `de|à`.

Figure 2.3 shows the two PTDs defined by the InterrogativeClauseObject class. The left one corresponds to total interrogative indirect clauses, whereas the right one corresponds to partial interrogative indirect clauses. The first ones require the complementizer *si*, which is expressed with the negative feature `cpl` ← `si`.

2.2 Predicate Complements

Predicate complements are complement that behave as predicates over the subject or the direct object of the verb on which they depend. Here are various examples of predicate complements[3].

(2.7) *L' entreprise reste un échec .*
The company remains a failure .
The company remains a failure.

(2.8) *Jean trouve Marie abandonnée par ses amis .*
Jean finds Marie abandoned by her friends .
Jean finds Marie abandoned by her friends.

(2.9) *Jean entend Marie chanter .*
Jean is hearing Marie singing .
Jean is hearing Marie singing.

(2.10) *Le problème est de rentrer tard .*
The problem is to come home late .
The problem is to come home late.

(2.11) *Marie passe pour une femme intelligente .*
Marie looks a woman clever .
Marie looks a clever woman.

(2.12) *Marie passe pour jalouse de Pierre .*
Marie is seen as jealous of Pierre .
Marie is seen as jealous of Pierre.

(2.13) *Marie passe pour être une femme intelligente .*
Marie looks to be a woman clever .
Marie is seen as being a clever woman.

The examples above illustrate different cases of predicate complements. Sentence (2.7) illustrates a predicate complement related to the subject *l'entreprise*, whereas Sentence (2.8) illustrates a predicate complement related to the object *Marie*. A basic

[3] Another analysis of Sentence (2.9) considers *Marie chanter* as a whole infinitive clause and *Marie* as the subject of this clause but it has some difficulty to express the relative independence of *Marie* with respect to *chanter* as in the sentence *Jean entend chanter Marie.*

class, PredicateCompl, expresses the common features of all cases. The different cases are distinguished according to the form of the complements and they give rise to different classes inheriting the PredicateCompl class:

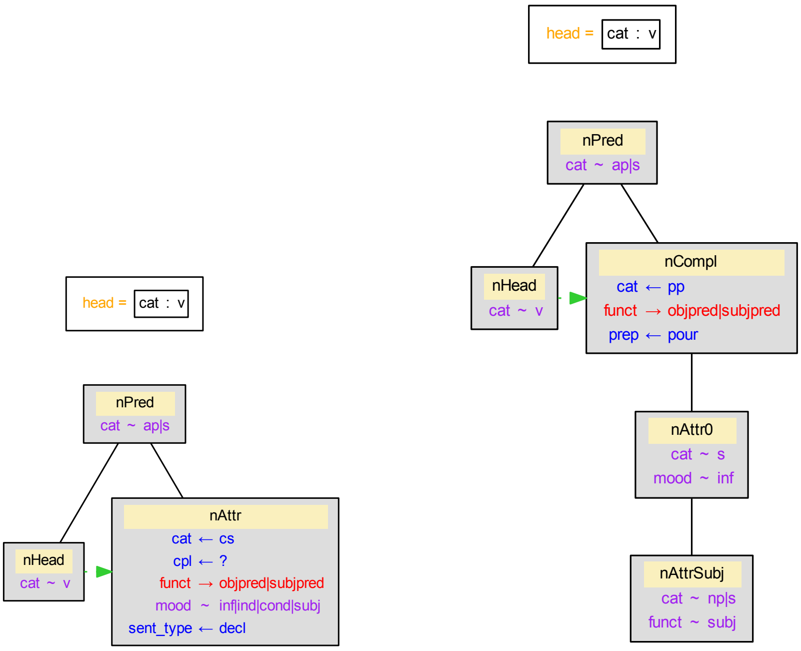

Figure 2.4: The PTDs defined by the DirectPredicateComplementedClause and IndirectPredicateInfinitiveClause classes

- DirectPredicateComplNounPhrase for predicate complements that are noun phrases (Sentence (2.7)),
- DirectPredicateComplAdjectivalPhrase for predicate complements that are adjectival phrases (Sentence (2.8)),
- DirectPredicateNonComplementedClause for predicate complements that are direct infinitives (Sentence (2.9)),
- DirectPredicateComplementedClause for predicate complements that are clauses introduced with a complementizer (Sentence (2.10)),
- IndirectPredicateComplNounPhrase for predicate complements that are noun phrases introduced with a preposition (Sentence (2.11)),

- IndirectPredicateComplAdjectivalPhrase for predicate complements that are adjectival phrases introduced with a preposition (Sentence (2.12)),
- IndirectPredicateInfinitiveClause for predicate complements that are infinitives introduced with a preposition (Sentence (2.13)).

For instance, Figure 2.4 shows the PTDs defined by the DirectPredicateComplementedClause and IndirectPredicateInfinitiveClause classes.

The right PTD includes particular nodes: node *nAttr0* representing the complement infinitive without its preposition and node *nAttrSubj* representing the subject of this infinitive. It will have the subject or the object of the verb as its antecedent according to the argument to which the predicate complement is related.

2.3 Indirect objects

Indirect objects are required complements introduced with a preposition, with the exception of some cases studied in the two previous sections. Here are various examples of indirect objects.

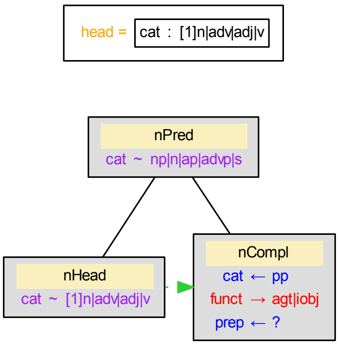

Figure 2.5: The PTD defined by the IndirectObject class

(2.14) *Marie est interrogée par Jean .*
Marie is asked by Jean .
Marie is asked by Jean.

(2.15) *Jean parle de Marie .*
Jean speaks about Marie .
Jean speaks about Marie.

(2.16) *Jean propose une collaboration avec le Japon .*
Jean proposes a collaboration with Japan .
Jean proposes a collaboration with Japan.

(2.17) *Jean est attentif au cours .*
Jean is attentive to the class .
Jean is attentive to the class.

(2.18) *Jean parvient à dormir .*
Jean succeeds in sleeping .
Jean succeeds in sleeping.

(2.19) *J'emène les enfants danser .*
I take the children dancing .
I take the children dancing.

(2.20) *Les bénéfices vont diminuant .*
The profits are going decreasing .
The profits are going decreasing.

(2.21) *Il a le souci de bien faire .*
He has the worry of well doing .
He is anxious to do well.

(2.22) *Il a le souci que la lettre arrive à Marie .*
He has the worry that the letter arrives to Marie .
He is anxious that the letter arrives to Marie.

Contrary to direct objects or predicate complements, indirect objects can be complements of verbs, nouns and adjectives, as the examples above illustrate it.

A basic class, IndirectObject, which inherits the PredComplement class, expresses the common features of all cases and it generates the PTD shown on Figure 2.5.

The different cases are distinguished according to the form of the complements and they give rise to different classes inheriting the IndirectObject class:

- AgentObject for agent complements (Sentence (2.14));
- NominalIndirectObject for other nominal complements (Sentences (2.15), (2.16) and (2.17));
- ClausalIndirectObject for indirect clausal complements introduced with a preposition (Sentence (2.18)); the complements can be infinitives or present participles.

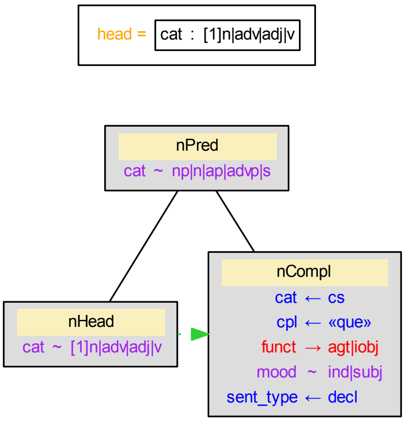

Figure 2.6: The PTD defined by the FiniteClauseDeObject class

Some complements are not introduced with prepositions, and nevertheless, they are considered as indirect objects because they verify tests for indirect objects and a preposition introducing them can be suggested. They are represented with the following classes inheriting the PredComplement class.

- InfinitiveIndirectObjectWithoutPreposition for indirect infinitive complements introduced without a preposition (Sentence (2.19))[4],
- PresentParticipleIndirectObjectWithoutPreposition for present participle complements which are indirect objects without preposition (Sentence (2.20)),
- FiniteClauseDeObject for finite clauses, introduced with the complementizer *que* and used as complements introduced with the preposition *de* (Sentence (2.22)). Figure 2.6 shows the PTD defined by this class.

[4] Even if there is no preposition, the complement is considered as indirect because the complement does not verify the usual tests for recognizing direct objects. A destination preposition is implicit.

Chapter 3

Verbs

3.1 Interfaces with the lexicon

Verbs are characterized in interfaces with the feature `head.cat = v`. Their morphological features and some syntactic properties are gathered in the `head` feature:

- `aff`: it takes the values `en`, `le`, `y` according to the affix that is associated with the verb (*en vouloir à*); if the verb takes no affix, the value of the feature is `voidaff`;
- `impers`: if the verb always takes an impersonal construction, the value of the feature is `always` (*falloir*); if the verb never takes an impersonal construction, the value of the feature is `never` (*comporter*); if the verb can enter personal as impersonal constructions, the value is `maybe` (*arriver, vendre*); for transitive verbs, the concerned impersonal construction is implicitly a passive construction;
- `mood`: it gives the mood of the verb, which can take the following values: `cond` (conditional), `imp` (imperative), `ind` (indicative), `inf` (infinitive), `pastp` (past participle), `presp` (present participle), `subj` (subjunctive);
- `num`: it gives the number of the verb: `pl` (plural) or `sg` (singular);
- `pers`: it gives the person of the verb: `1`, `2` or `3`;
- `pronominal`: if the verb is essentially pronominal, the value of the feature is `always` (*s'enfuir*); if the verb never enters a pronominal construction, the value of the feature is `never` (*venir*); if the verb enters a pronominal construction by accident, the value is `maybe` (*laver, rencontrer*);
- `tense`: it gives the tense of the verb: `fut` (future), `imperf` (imperfect), `pres` (present), `past`;
- `trans`: it can takes the values `true` or `false` depending on whether there is a transitive entry in the lexicon with the same lemma; this feature is used for past participles that are heads of adjectival phrases to recognise if the voice is active or passive;

- `verb_type`: its possible values are `aux`, `caus`, `modal`, `presentative`, `standard`, depending on whether the verb is a tense auxiliary, a causative auxiliary, a modal auxiliary, a presentative or a standard verb.

The subcategorisation frame of a verb is described with features which are put in parallel with the head feature and describe the required syntactic arguments of the verb:

- `caus`: the complement verb of causative auxiliaries
- `iobj1`: first indirect object,
- `iobj2`: second indirect object,
- `iobj3`: third indirect object,
- `obj`: direct object,
- `obj_modal`: infinitive object of a modal auxiliary,
- `objpred`: object predicate complement, a predicate complement that agrees with the object of the verb,
- `subj`: subject,
- `subjpred`: subject predicate complement, a predicate complement that agrees with the subject of the verb,

For each argument feature, the properties of the argument are described with the following features:

- `cat`: the category of the argument, which can take the values `ap` (adjective phrase), `np` (noun phrase), `pp` (prepositional phrase) or `s` (sentence); if the argument is introduced with a complementizer or a preposition, the considered category is that of the argument without the complementizer or the preposition; the presence of these link words is indicated with another feature; for instance, one of the lexical entries for the verb *aller* has a feature `obj1.cat = np` to describe the locative complement of the verb;
- `funct`: the syntactic function of the argument, which can take the values `iobj` (indirect object)[1], `modal` (object of a modal auxiliary), `obj` (direct object), `objpred` (object predicate complement), `subj` (subject), `subjpred` (subject predicate complement);

[1] An indirect object is an object introduced with a preposition but in some cases, the preposition may be missing as in the sentence *Il emmène Marie chercher son fils (he takes Marie getting her son)*, where *chercher son fils* is regarded as an indirect object of *emmène* without preposition.

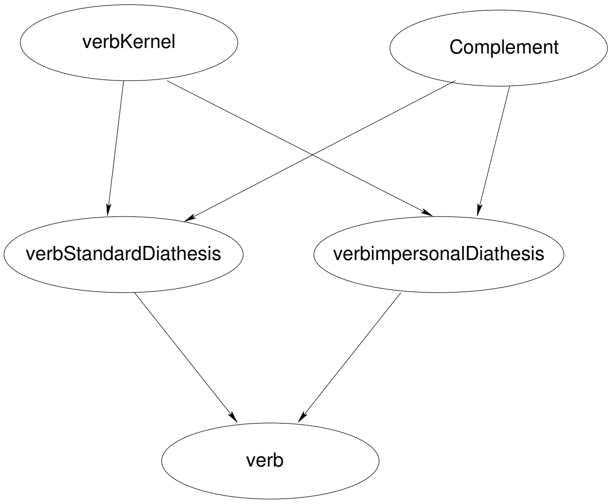

Figure 3.1: The dependencies between the modules of the verb grammar

- `control`: if the argument is an infinitive, the function of the other argument that is its subject; the feature can take the values `iobj`, `obj`, `subj` or `void`, if there is no control over the infinitive; for instance, one the lexical entries for the verb *permettre (allow)* has the feature `obj.control = iobj` because it takes an infinitive as its direct object and the subject of this infinitive is the indirect object of *permettre*;

- `cpl`: a possible complementizer if the argument is a sentence; if the argument is a sentence that requires no complementizer, it can be indicated with the feature `cpl = voidcpl`;

- `mood`: the possible mood of the argument if it is a sentence;

- `prep`: the possible preposition introducing the argument; if the argument requires no preposition, it is possible to indicate it with the feature value `voidprep`;

- `sem`: the semantic type of the argument, which can take the values `abstr` (abstract), `anim` (animate), `inanim` (inanimate) or `void` if the argument has no semantic content;

- `sent_type`: the type of the sentence, if the argument is a sentence; this type can be `decl` (declarative), `excl` (exclamatory), `imper` (imperative) or `inter` (interrogative);

The two first features above are always present in the description of an argument; the other ones are optional.

3.2 The verb modules

The verb grammar is the most important part of FRIGRAM. it includes about 2600 EPTDs. Because of its size, it is shared out in several modules, which depend on each other:

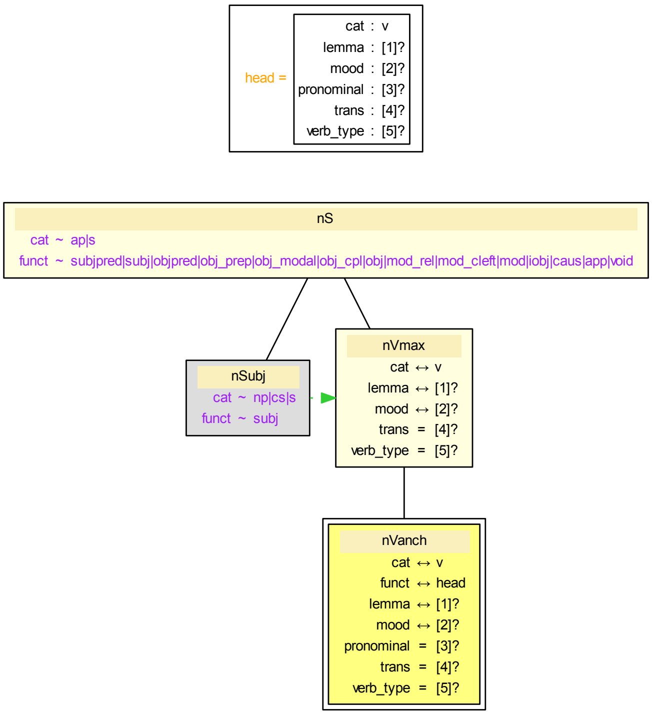

Figure 3.2: EPTD defined by the BasicVerb class.

- *verbKernel* concerns the common kernel of all verbs, auxiliaries included. It describes the verb without considering its complement, that is the following aspects and their mutual dependencies: the morphology of the verb, its mood, the voice of the clause it governs, the syntactic function of this clause in its environment and finally the shape of the subject. The syntactic function of the clause in its environment can be the whole sentence, the subject or a complement of another verb or a participial phrase attributive of a noun.

- *verbStandardDiatheses* describes the usual verb diatheses: active, passive and middle[2]. The classes of this module are built by conjunctive combination of classes coming from modules *verbKernel* and *Complement*.

[2] Causative constructions are not considered in the same way as active and passive diatheses.

- *verbImpersonalDiathesis* describes the impersonal verb diathesis, that is the diathesis in which the subject is impersonal. It uses the same mechanism of combination as the previous module.

- *verb*, which describes the different verb families according to their subcategorization frames. The classes of the module are built by disjunctive combination of classes coming from the two modules related to the verb diatheses.

The dependencies between the modules are represented on the diagram of Figure 3.1.

3.3 The verb kernel or the verb without its complements

The BasicVerb class describes the common skeleton to all EPTDs anchored by verbs. It is presented in Figure 3.2 with the following nodes:

- *nVanch* represents the bare verb,

- *nVmax* represents the verbal kernel constituted of the bare verb with its possible affixes, clitic pronouns and adverbs; it is the mother node of *nVanch*;

- *nS* represents the clause or the participial phrase that includes the verbal kernel as an immediate sub-constituent;

- *nSubj* represents the subject of the verb and it is a sister node of *nVmax*. *nSubj* precedes *nVmax*, even with subject inversion: in this case, *nSubj* represents a trace of the actual subject, which is put after *nVmax*.

Node *nVmax* share a saturated `mood` feature with *nVanch* but not necessarily with *nS*. It depends on whether *nVanch* determines the inflection of *nS* or not.

3.3.1 Inflectional versus non inflectional verb

The BasicVerb class is divided into two subclasses, ActiveInflectionVerb and CompoundVerb, according to the role of *nVanch* in the determination of the inflection of *nS*.

- For ActiveInflectionVerb, the anchoring verb determines the mood of *nS*, and as a consequence its syntactic function; it also determines the shape of *nSubj*.

- For CompoundVerb, the verb is a past participle combined with tense or passive auxiliaries and the inflection of *nS* is determined by one of these auxiliaries, according to its position and mood.

Causative auxiliaries are regarded as full verbs taking specific complements. They contribute to a variant of the active diathesis.

The verb that gives the inflection to *nS* is called the *inflectional verb* and its corresponding *nVmax* node is called the inflectional daughter of *nS*. In this function, it is named *nInfl*. The inflectional node is not always the head of *nS*. This is only the case when the head is a simple verb. If the verb is a past participle composed with auxiliaries, it is not the inflectional verb of *nS* but it is the head.

In the following examples, verbs that are heads of a clause or a participial phrase are in bold with a subscriptH, those that are inflectional verbs are in bold with a subscript I.

(3.1) *Jean* ***vient***$_{HI}$ *aujourd'hui .*
Jean is coming today .
Jean is coming today.

(3.2) *Jean veut* ***venir***$_{HI}$ *aujourd'hui .*
Jean wants to come today .
Jean wants to come today.

(3.3) *Jean* ***est***$_{I}$ ***venu***$_{H}$ *aujourd'hui .*
Jean came today .
Jean came today.

(3.4) *Jean* ***a***$_{I}$ ***fait***$_{H}$ ***venir***$_{HI}$ *Marie aujourd'hui .*
Jean got to come Marie today .
Jean got Marie to come today.

(3.5) *Jean* ***a***$_{I}$ *été* ***invité***$_{H}$ *par le directeur .*
Jean was invited by the director .
Jean was invited by the director.

(3.6) *Jean* ***ayant***$_{I}$ *déjà* ***rencontré***$_{H}$ *Marie ne viendra pas à la*
Jean having already met Marie will not come to the
réunion .
meeting .
Jean having already met Marie will not come to the meeting.

(3.7) *Jean* ***ayant***$_{I}$ *déjà* ***rencontré***$_{H}$ *Marie , celle-ci ne viendra pas*
Jean having already met Marie , this one will not come
à la réunion .
to the meeting .
Jean having already met Marie, this one will not come to the meeting.

(3.8) *Jean croit Marie* ***abandonnée***$_{HI}$ *par Pierre .*
Jean believes Marie abandoned by Pierre .
Jean believes Marie to be abandoned by Pierre.

In Examples (3.1), (3.3) and (3.5), the *nS* node from the EPTD of the verbs in bold corresponds to the whole sentence.

In Example (3.2), it corresponds to the infinitive clause *venir aujourd'hui*.

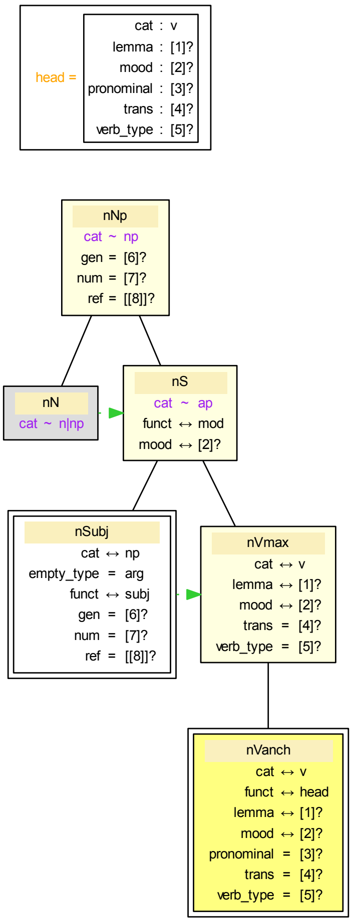

Figure 3.3: EPTD defined by the AttributiveAdjectivalParticiple class.

In Example (3.4), the causative auxiliary *fait* is regarded as taking two complements, the infinitive *venir* and the direct object *Marie*. The *nS* node from the EPTDs of *a* and *fait* represents the whole sentence but for *venir*, it reduces to this verb.

In Sentences (3.6) and (3.8), *nS* represents the respective participial phrases *ayant déjà rencontré Marie* and *abandonnée par Pierre*.

Sentence (3.7) differs from Sentence (3.6) because *nS* represents the phrases *Jean ayant déjà rencontré Marie*, which is regarded as a standard clause.

According to the syntactic function of *nS*, the ACTIVEINFLECTIONVERB class is refined in two kind of subclasses:

- when *nS* represents a participial phrase used an adjectival phrase, it is refined in the ATTRIBUTIVEADJECTIVALPARTICIPLE and PREDICATEADJECTIVALPARTICIPLE classes,
- when *nS* represents a clause, the class is refined in the FINITEVERB, INFINITIVEVERB and CLAUSEHEADPARTICIPLE classes, according to the mood of the anchor verb.

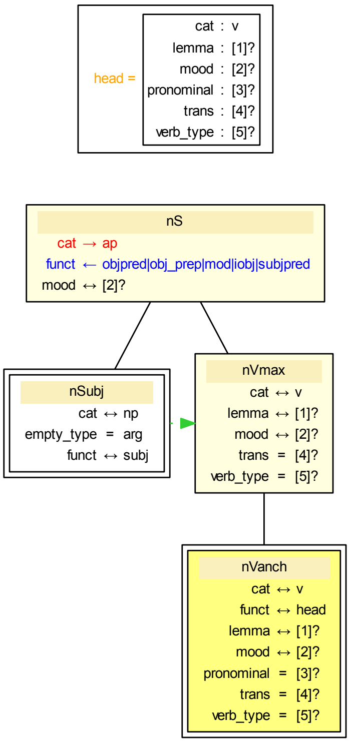

Figure 3.4: EPTD defined by the PREDICATEADJECTIVALPARTICIPLE

3.3.2 Verbs contributing to the inflection of participial phrases

The ATTRIBUTIVEADJECTIVALPARTICIPLE and PREDICATEADJECTIVALPARTICIPLE classes define two EPTDs anchored by verbs determining the inflection of participial phrases playing the role of adjectival phrases. These EPTDs, which are respectively represented with Figure 3.3 and 3.4 correspond to two functions of the participles: attributive or predicate complement.

In both EPTDs, node *nS* represents the participial phrase. In Examples (3.6) and (3.8), which illustrate them, this participial phrase is respectively instantiated with *ayant déjà rencontré Marie* and *abandonnée par Pierre*. Participial phrases are considered as complete clauses with an empty node *nSubj* being the subject of the clause.

The verb anchoring each EPTD contributes to the inflection of the participial phrase. This is expressed by sharing the `mood` feature between *nS* and *nVmax* nodes. For Example (3.6), the verb anchor of the EPTD is *ayant* and the shared feature is `mood = presp`. For Example (3.8), the verb anchor of the EPTD is *abandonnée* and the shared feature is `mood = pastp`.

Now, let us examine the differences between the two EPTDs. The EPTD of Figure 3.3 corresponds to the attributive function of the participial phrase, as illustrated with Example (3.6). The *nS* node carries two saturated features `cat` ↔ `ap` and `funct` ↔ `mod`, because it is a modifier of a noun represented with node *nN*, *nNp* representing the resulting noun phrase. In Example (3.6), nodes *nN* and *nNp* are respectively instantiated with *Jean* and *Jean ayant déjà rencontré Marie*.

The EPTD of Figure 3.4 corresponds to the predicate function of the participial phrase, as illustrated with Example (3.8). Node *nS* carries a positive feature `cat` → `ap` and a negative feature `funct` ← `objpred|obj_prep|mod|iobj|subjpred`, because it can provide an adjectival phrase as a predicate complement to an appropriate constituent that assigns it one of the functions given by the previous disjunction. In Example (3.8), node *nS* takes the `objpred` function from the verb *croit*.

The AttributiveAdjectivalParticiple class is divided into two subclasses: AttributiveAdjectivalpresentParticiple and AttributiveAdjectivalPastParticiple. The first class concerns present participles and the second class concerns past participles. This differentiation is necessary because in the second case, the agreement in gender and number must be explicitly described.

There is a similar division of the PredicateAdjectivalParticiple class into the PredicateAdjectivalPresentParticiple and PredicateAdjectivalPastParticiple classes. In Example (3.8), the past participle *abandonnée* anchors the EPTD of the PredicateAdjectivalPastParticiple class, which achieves the agreement between the past participle and its subject and the EPTD anchored by *croit* achieves the agreement between this subject and the direct object *Marie*.

3.3.3 Verbs contributing to the inflection of standard clauses

Apart from participial phrases, the verbs anchoring the ActiveInflectionVerb class contribute to the inflection of standard clauses. According to its mood, the ActiveInflectionVerb is refined in the following subclasses: FiniteVerb for the finite moods, InfinitiveVerb for the infinitive mood, ClauseHeadParticiple for the present and past participles heads of clauses.

The classes define the EPTDs presented in Figures 3.5 and 3.6. The inflectional property of the verb is expressed with the `mood` feature which is shared by the *nS*, *nVmax* and *nVanch* nodes and saturated in the three nodes. It implies the syntactic function of the clause represented by the *nS* node.

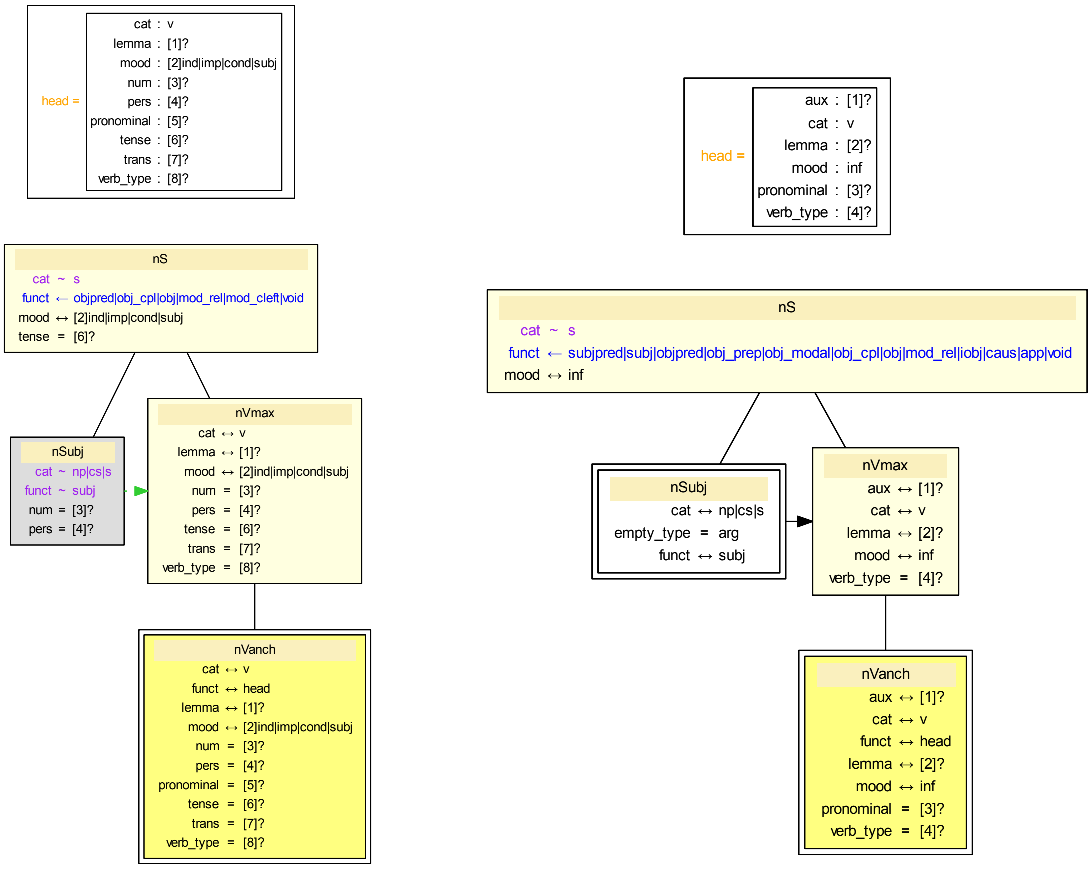

Figure 3.5: EPTDs defined by the FiniteVerb and InfinitiveVerb classes

Now, every class has its specificities. For the FiniteVerb class presented in Figure 3.5, a `tense` feature is added to the *nS*, *nVmax* and *nVanch* nodes with a shared value. It is illustrated with Examples (3.1), (3.3), (3.4) and (3.5). This class is divided in two subclasses according to whether the mood is imperative or not: ImperativeVerb and NonImperativeFiniteVerb. For the first one, the subject is empty and its `cat` and `funct` features are saturated because an imperative verb requires no external subject.

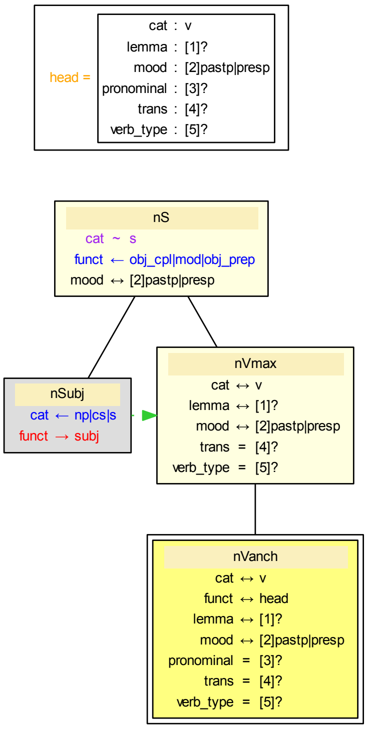

Figure 3.6: EPTD defined by the PARTICIPLEVERB class

The INFINITIVEVERB class, presented in Figure 3.5, is illustrated with Examples (3.2) and (3.4). Its particularity lies in the form the subject which is empty and for which the `cat` and `funct` features are saturated because an infinitive verb requires no external subject.

Finally, the PARTICIPLEVERB is illustrated with Example (3.7). The main difference with respect to the previous classes lies in the polarities attached at the *nSubj* node, which express that an external subject is always required. This class is divided in two subclasses according to the mood of the participle: PRESENTPARTICIPLEVERB and PASTPARTICIPLEVERB. For the PASTPARTICIPLEVERB class, the subject *nSubj* agrees with the verb anchored at *nVanch* in number and person. The two features are also raised to *nVmax*

The different classes expressing different forms of clause inflectional verbs are gathered by disjunctive composition in a unique class ACTIVEINFLECTIONCLAUSEVERB, except PASTPARTICIPLEVERB because of its particular behaviour: it excludes combination with a reflexive pronoun. In fact, ACTIVEINFLECTIONCLAUSEVERB is the disjunction of IMPERATIVEVERB, NONIMPERATIVEVERB, FINITEVERB, INFINITIVEVERB and CLAUSEHEADPRESENTPARTICIPLE.

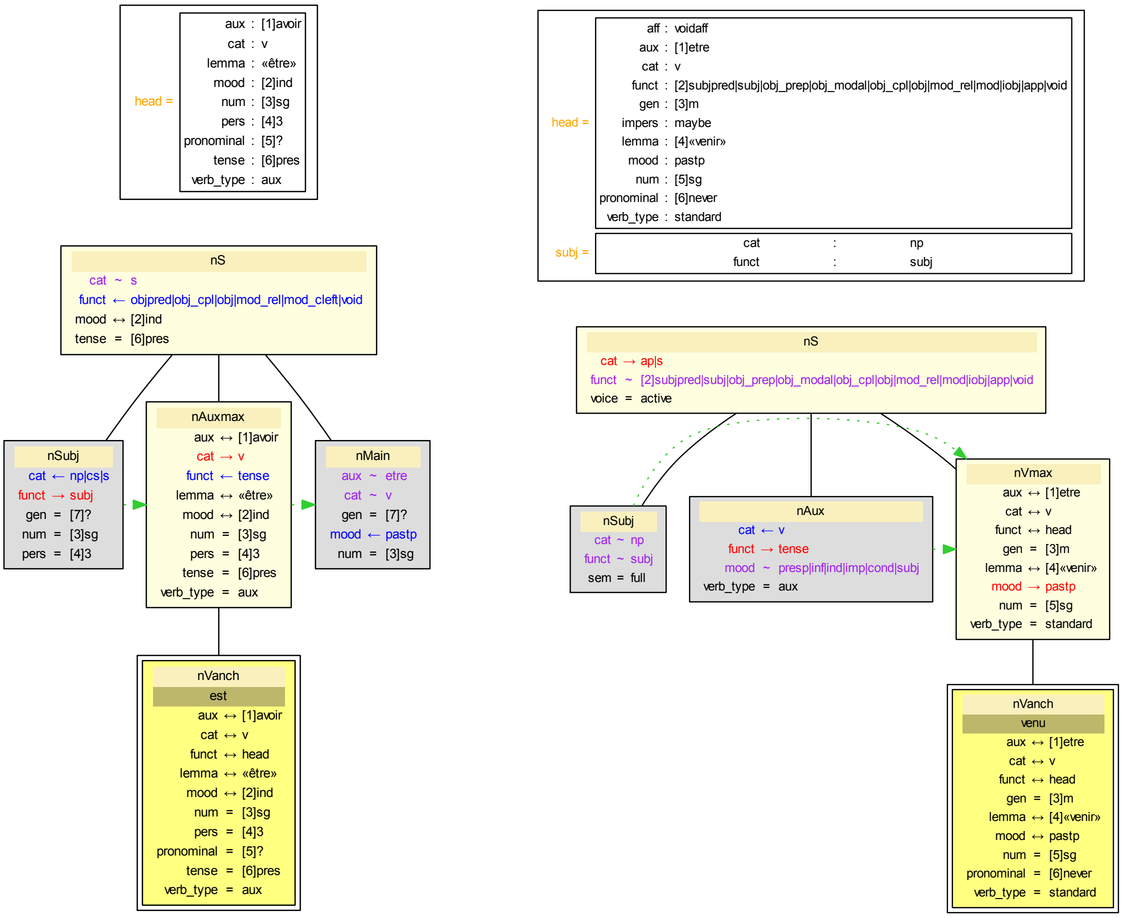

Figure 3.7: EPTDs used for *est* and *venu* in the parsing of *Jean est venu aujourd'hui.*

3.3.4 Past participles combined with auxiliaries to build compound verbs

The interaction between a past participle and an auxiliary is performed by four kinds of features: `cat`, `funct`, `mood` and `verb_type`.

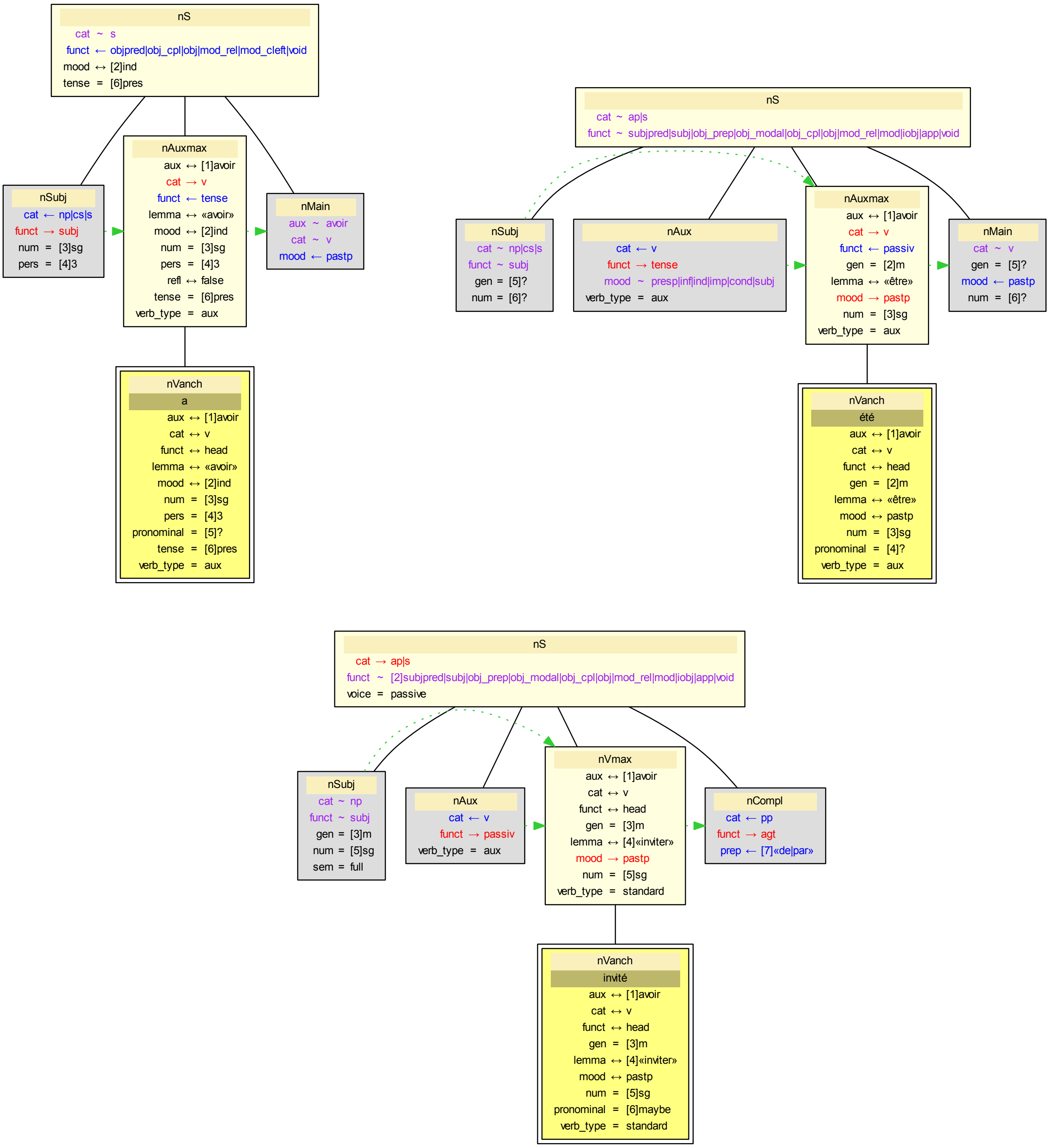

Figure 3.8: EPTDs associated with the verbs *a*, *été* and *invité* to compose the sentence *Jean a été invité par le director (Jean was invited by the director).*

Figure 3.7 illustrates this game between the `cat`, `funct`, `mood` and `verb_type` features and the polarities of the EPTDs anchored by the past participle *venu* and the auxiliary *est* in the composition of the sentence *Jean est venu aujourd'hui (Jean came today).*

On this figure, the left EPTD is attached at the auxiliary. Nodes *nAuxmax* and *nMain* are aimed to be respectively headed by the auxiliary and the past participle. They will be merged with the corresponding *nAux* and *nVmax* nodes of the right EPTD attached at the past participle. The auxiliary and the past participle are in separate daughters of *nS*, because an adjunct of the sentence may be inserted between a verb and its auxiliary, as in the following sentence: *Jean a, cet après-midi, rencontré Marie. (Jean, this afternoon, met Marie)*

Node *nAuxmax* carries features `cat` → `v`, `funct` ← `tense` and `verb_type = aux` to express that *est* is an available auxiliary which expects a function of tense auxiliary. It will merge with node *nAux* of the EPTD associated with *venu*, which carries dual features.

At the same time, node *nMain* carries a `mood` ← `pastp` feature to express that *est* expects a past participle. It will merge with node *nVmax* the EPTD associated with *venu*, which carries a dual feature.

In the same sentence, a verb can have both roles: auxiliary and main verb. This the case for Sentence (3.5), in which *été* is a passive auxiliary for *invité* and at the same time, it is a main verb with *a* as its tense auxiliary.

Figure 3.8 shows the EPTDs used for the three verbs in the composition of the sentence *Jean a été invité par le directeur*[3].

In the grammar, the interaction between auxiliaries and main verbs is expressed within two classes: Auxiliary and CompoundVerb

The past participle side

The CompoundVerb class expresses that the anchoring verb is a past participle expecting an auxiliary. It defines the EPTD for which the last example of Figure 3.8 gives an instantiation.

The CompoundVerb class is divided into two subclasses, according to the different functions of auxiliaries: TenseCompoundVerb and PassiveCompoundVerb.

For the two classes, the verb is a past participle with two agreement features: `gen` (gender) and `num` (number). The PassiveCompoundVerb class expresses a systematic agreement between the past participle and the subject. For the TenseCompoundVerb class, the agreement rule is very complex and depends on the transitivity of the verb and the position of the object with respect to the verb. We have partially left this problem aside in the current version of the grammar. The problem is solved in the case

[3] The example shows why the `cat` feature of *nAuxmax* node is positive for auxiliaries and the `cat` feature of *nVmax* node is saturated for main verbs that are not auxiliaries. If this feature would be positive in all cases and if the `cat` feature would be negative for all *nMain* nodes in the EPTDs of auxiliaries, the parsing of the sentence would fail: the `cat` feature for *été* must neutralise two negative features coming from *a* and *invité*, which is not possible.

on intransitive verbs with *être* as their tense auxiliary. In this case, the past participle agrees with the subject.

The auxiliary side

The Auxiliary class defines the common skeleton of the tense and passive auxiliary PTDs, which is shown in Figure 3.9 and offers an exact duality of polarities for its *nAuxmax* and *nMain* nodes with respect to the corresponding nodes *nAux* and *nVmax* of the CompoundVerb class.

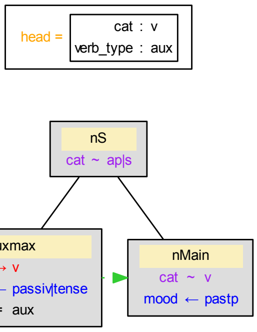

Figure 3.9: The PTD defined by the Auxiliary class.

Node *Auxmax* represents the verbal kernel anchored with the auxiliary and node *nMain* represents the verbal kernel of the expected main verb that combines with the auxiliary. The Auxiliary class is then divided into two subclasses: Vavoir_V1pastp and Vetre_V1pastp.

The Vavoir_V1pastp class inherits the Auxiliary class and one of the three following classes:

- ActiveInflectionClauseVerb, when the *avoir* auxiliary contributes to the inflection of a clause, as Sentences (3.4), (3.5) and (3.7) illustrate it; Figure 3.8 shows one the 4 EPTDs generated by the class; each EPTD corresponds to a particular mood of the auxiliary;

- AttributiveAdjectivalPresentParticiple defining an EPTD for *avoir* auxiliary contributing to the inflection of a present participial phrase with an attributive function, such as in Sentence (3.6);

- PredicateAdjectivalPresentParticiple defining an EPTD for *avoir* auxiliary contributing to the inflection of a present participial phrase used in another function than attributive, such as in Sentence (3.7).

The Vetre_V1pastp class represents the use of *être* as a tense or passive auxiliary. It inherits the Auxiliary class and one of the four classes: ActiveInflectionClauseVerb, AttributiveAdjectivalPresentParticiple, PredicateAdjectivalPresentParticiple or TenseCompoundVerb. Unlike *avoir*, the last class is justified by the use of *être* as passive auxiliary in past participle. The EPTD corresponding to the case is shown on Figure 3.8 and it is used in Sentence (3.4).

The use of *être* as tense auxiliary is made more complicated because it is used in all reflexive constructions, which will be studied in details now.

3.3.5 The reflexive constructions

Reflexive pronouns are considered as clitic pronouns and they are are used in very different contexts. Here are examples of these different contexts. In each example, the inflectional verb is a tense auxiliary (in bold), to illustrate the influence of reflexive pronouns on the choice of the tense auxiliaries.

(3.9) *Jean* ***a*** *acheté une voiture .*
Jean has bought a car .
Jean has bought a car.

(3.10) *Jean s’* ***est*** *acheté une voiture .*
Jean himself has bought a car .
Jean has bought himself a car.

(3.11) *Ils se* ***sont*** *rencontrés hier .*
they themselves are met yesterday .
they met yesterday.

(3.12) *Jean s’**est** beaucoup ennuyé à Paris .*
Jean was very bored in Paris .
Jean was very bored in Paris.

(3.13) *Ces vins se* ***sont*** *bien vendus .*
These wines were well sold .
These wines were well sold.

There are four possible contexts for reflexive pronouns: essentially pronominal verbs (Example (3.12)), transitive verbs with a reflexive object (Example (3.11)), verbs with a dative reflexive complement (Example (3.10)), middle voice (Example (3.13)).

The addition of a reflexive pronoun to the verbal kernel is controlled by the polarised feature `refl`, which can take three values:

`aff`: the reflexive pronoun is an affix representing no argument of the verb for essentially pronominal verbs and the middle voice; in both cases, a negative feature `refl` ← `aff` is added by the responsible verb to the kernel of the inflectional verb, which is different of the first one for a compound tense;

- `arg`: the reflexive pronoun represents a direct or an indirect object of the verb; in this case, the EPTD of the reflexive pronoun add a saturated feature `refl` ↔ `arg` to the kernel of the inflectional verb that it modifies;
- `false`: there is no reflexive pronoun as in Example (3.9).

In any cases, in a compound tense with a reflexive pronoun, the tense auxiliary is *être*. Figure 3.10 shows how this constraint is implemented in the EPTDs for the tense auxiliaries *a* and *est*, the second one being used with a reflexive pronoun.

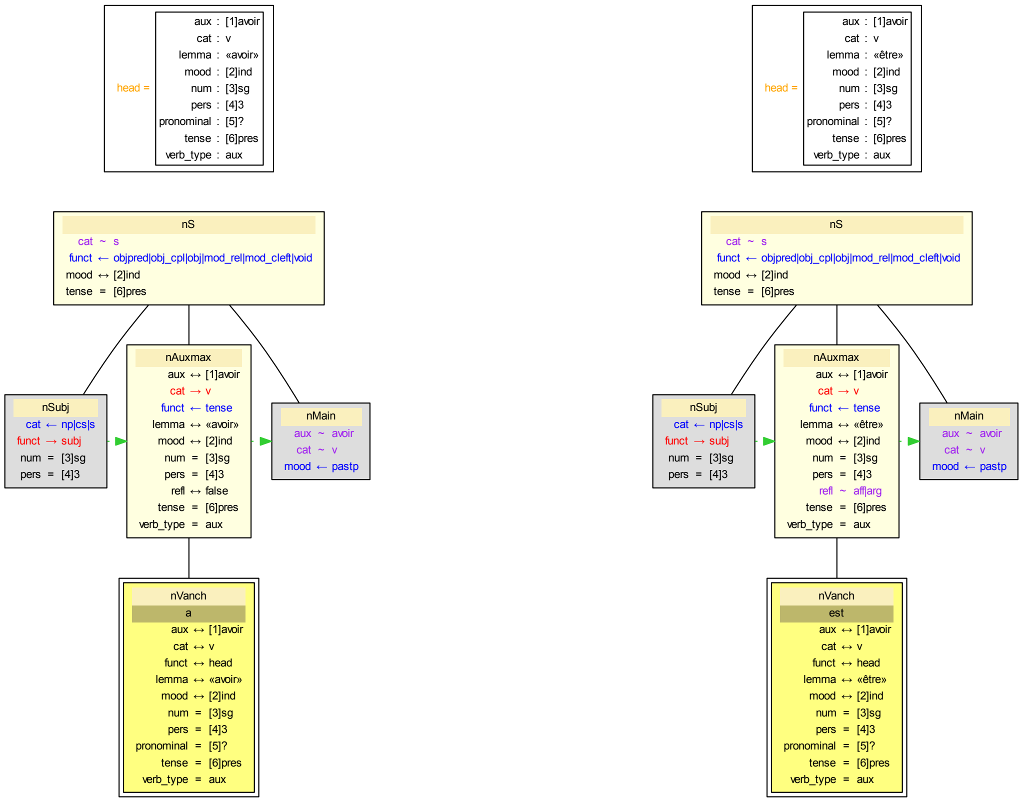

Figure 3.10: EPTDs for the tense auxiliaries *a* and *est*.

The EPTD of the tense auxiliary *avoir* imposes the feature `refl` ↔ `false` to its verbal kernel, so that the parsing of *ils s'ont rencontrés* fails. The EPTD of the tense auxiliary *être* represents its use with a verb requiring *avoir* in a reflexive context (Examples (3.10), (3.11) and (3.13)). The EPTD verifies with a virtual feature `refl` ∼ `aff|arg` that the verbal kernel of *est* has a feature `refl` with the value `arg` or `aff`. In this way, the parsing of *Jean est acheté une voiture* fails.

3.3.6 The different voices of full verbs

Full verbs have three voices, active, passive and middle. Causative is not considered as a specific voice and correlatively, causative verbs are not considered as auxiliaries but as full verbs. Causative constructions will be studied with the different diatheses of full verbs.

The four first examples of the previous subsection illustrate the active voice of verbs in a compound tense. The last example illustrates the middle voice. Here are additional examples (the concerned verbs are in bold).

(3.14) *Jean a fait* ***venir*** *Marie aujourd'hui .*
Jean made come Marie today .
Jean made Marie come today.

(3.15) *Jean ayant déjà* ***rencontré*** *Marie ne viendra pas à la*
Jean having already met Marie do not come to the
réunion .
meeting .
Jean having already met Marie do not come to the meeting.

(3.16) *Le voleur* ***évanoui*** *dans la nature sera difficile à retrouver .*
The robber vanished in the nature will be difficult to find again .
The vanished robber will be difficult to find again.

(3.17) *Jean connaît la femme* ***invitée*** *par le directeur .*
Jean knows the woman invited by the director .
Jean knows the woman invited by the director.

(3.18) *Jean est* ***invité*** *par le directeur .*
Jean is invited by the director .
Jean is invited by the director.

(3.19) *Jean croit Marie* ***abandonnée*** *par Pierre .*
Jean believes Marie abandoned by Pierre .
Jean believes Marie to be abandoned by Pierre.

(3.20) *Le voyage* ***annulé*** *, nous avons du temps .*
The travel cancelled , we have time .
The travel being cancelled, we have time.

The three first sentences concern the active voice and the four last ones concern the passive voice. The three voices give rise to three classes: ActiveMorphology, PassiveMorphology and MiddleMorphology.

Active voice

The ActiveMorphology class is the disjunctive composition of two classes: NonReflexiveActiveMorphology and ReflexiveActiveMorphology. The distinction concerns the requirement or not of a reflexive pronoun with the verb. In case that a reflexive pronoun is requirement, it is the affix associated with an essentially pronominal verb. The requirement is expressed with a negative feature `refl` ← `aff` attached at node *nVmax*.

The NonReflexiveActiveMorphology class is illustrated with Sentences (3.14), (3.15) and (3.16). It is the disjunctive composition of classes that describe the different syntactic functions of the verb:

- ActiveInflectionClauseVerb, AttributiveAdjectivalPresentParticiple and PredicateAdjectivalPresentParticiple, for which the verb contributes to the inflection of the clause or the participial phrase; in these cases, the verb is not essentially pronominal;

- AdjectivalPastParticiple and ClauseHeadPastParticiple for past participles heads of adjectival phrases or sentences; in these cases, the verb may be an essentially pronominal verb but the reflexive affix is not present as in Sentence (3.16);

- TenseCompoundVerb, when the verb is a past participle composed with a tense auxiliary.

The ReflexiveActiveMorphology class concerns an essentially pronominal verb in the active voice with a reflexive pronoun. It is illustrated with Sentences (3.12) and (3.21). The class is defined from a disjunction of the ActiveInflectionClauseVerb, AttributiveAdjectivalPresentParticiple, PredicateAdjectivalPresentParticiple and ReflexiveTenseCompoundVerb classes.

Some verbs are used in the active voice with a clitic pronoun *le*, *en* or *y*, which represents no complement of the verb but affects its meaning, like *connaît* in Example (3.21).

(3.21) *Jean s'y* ***connaît*** *dans les capitales du monde .*
Jean knows in the capitals of the world .
Jean knows all about the capitals of the world.

The EPTD used for *connaît* to parse this example is given by Figure 3.11. In this EPTD, the node *nAff* represents the expected clitic *y*.

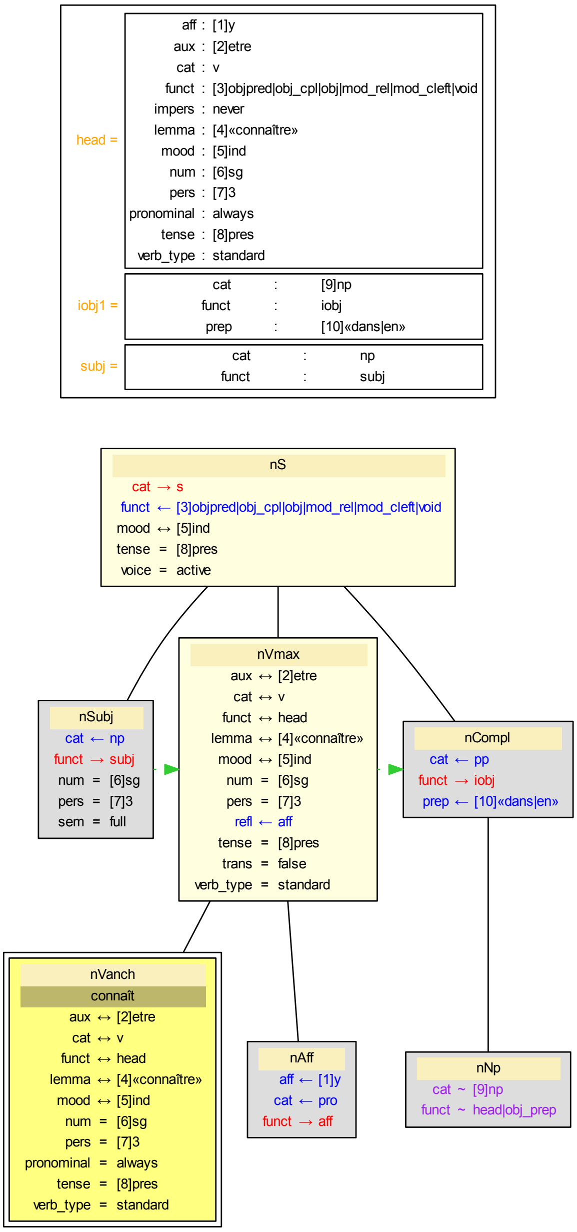

Figure 3.11: The EPTD used for *connaît* in the parsing of Sentence (3.21).

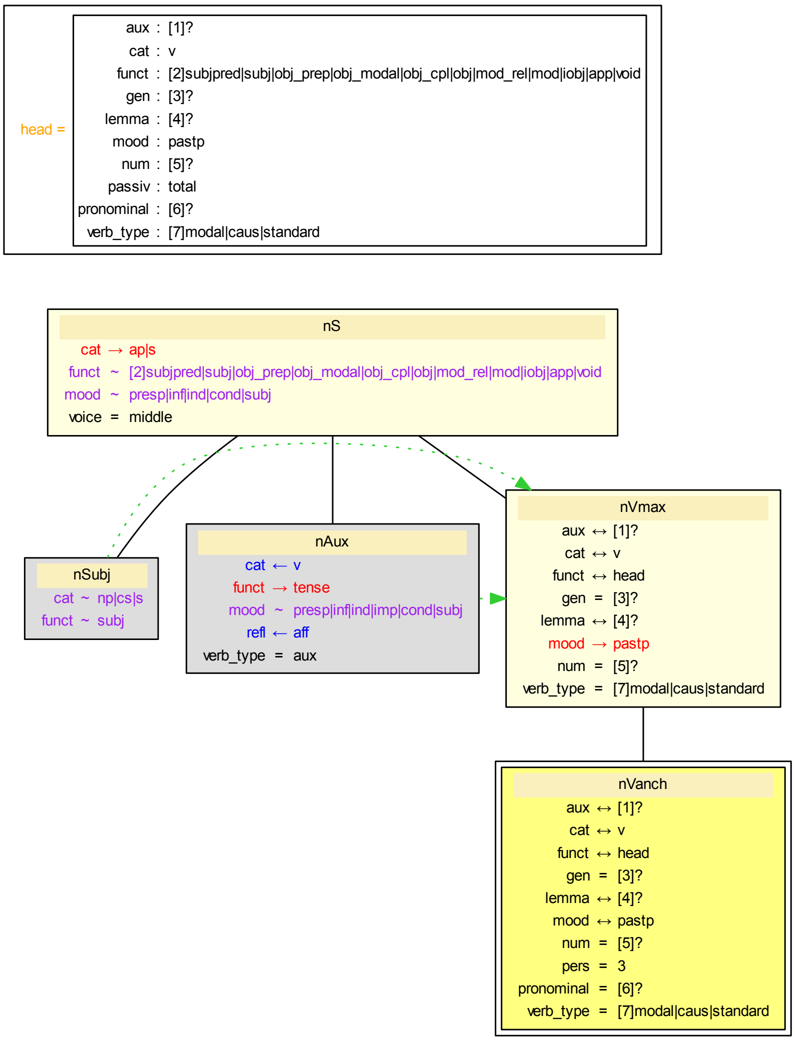

Figure 3.12: An EPTD defined by the MiddleMorphology class for finite verbs.

For almost each verb EPTD of FRIGRAM, there is two versions, a version with affix and a version without affix, which doubles the size of the verb grammar.

Passive voice

The PassiveMorphology class is the disjunctive composition of the following classes:

- AdjectivalPastParticiple for past participles heads of attributive participial phrases, such as in Sentences (3.17) and (3.19);

- PassiveCompoundVerb for past participles combined with a passive auxiliary, such as in Sentence (3.18);
- ClauseHeadPastParticiple for past participles heads of participial phrases complements of verbs, such as in Sentence (3.20).

Middle voice

For some transitive verbs, the middle voice represents a construction where the object of the active voice becomes the subject of the middle voice and at the same time the verb takes a reflexive pronoun. The MiddleMorphology class is the disjunction of classes that model the different cases: ActiveInflectionClauseVerb, AttributiveAdjectivalPresentParticiple and TenseCompoundVerb. Sentence (3.13) illustrates the case of the *TenseCompoundVerb* class, which defines the EPTD given by Figure 3.12. In this EPTD, node *nAux* represents the expected tense auxiliary and the negative feature `refl` ← `aff` means that the auxiliary requires the adjunction of a reflexive clitic.

3.4 The different verb diatheses

We consider the active, passive and middle diatheses. Used in a personal construction they are gathered in the VerbPersonalDiatheses module. Used in an impersonal construction, they are gathered in the VerbImpersonalDiatheses module.

Here are examples illustrating the different diatheses.

(3.22) *Jean* **vient** *aujourd'hui* *.*
Jean is-coming today .
Jean is coming today.

(3.23) *Que* *Jean* *ne* *vienne* *pas* **gêne** *Marie* *.*
that Jean does not come disturbs Marie .
Jean not coming disturbs Marie.

(3.24) *Jean* *est* ***invité*** *par* *le* *directeur* *.*
Jean is invited by the director .
Jean is invited by the director.

(3.25) *Partir* *est* *sérieusement* **envisagé** *.*
To go is seriously envisaged .
To go is seriously envisaged.

(3.26) *Ce* *vin* *se* **boit** *très* *frais* *.*
That wine is drunk very fresh .
That wine must be drunk very fresh.

(3.27) *Que Jean parte se* **comprend** *facilement .*
That Jean is leaving is understood easily .

Jean leaving can be understood easily.

(3.28) *Il* **pleut** *aujourd'hui .*
it is raining today .

it is raining today.

(3.29) *Il a été* **vendu** *beaucoup de voitures .*
it has been sold a lot of cars .

it has been sold a-lot of cars.

(3.30) *Il se* **vend** *beaucoup de voitures .*
It itself sells a lot of cars .

A lot of cars are sold.

(3.31) *Il a été* **décidé** *que Jean parte .*
It was decided that Jean go .

It was decided that Jean go.

Examples (3.22) and (3.23) illustrate the personal active diathesis. Examples (3.24) and (3.25) illustrate the personal passive diathesis and Examples (3.26) and (3.27) the personal middle diathesis. Examples (3.28), (3.29), (3.30) and (3.31) illustrate the different impersonal diatheses.

3.4.1 The VerbPersonalDiatheses Module

The personal active diathesis

The personal active diathesis is described by 38 classes according to the category of the subject and the required complements.

The two following classes are the ground classes of this family:

- NP_Vactive, which represents the active diathesis with a nominal personal subject; it inherits the ActiveMorphology class;
- S_Vactive, which represents the active diathesis with a clausal personal subject; it inherits the ActiveMorphology class.

Figure 3.13 represents the EPTDs defined by these classes for the conditional, indicative and subjunctive moods.

The right EPTD concerns verbs taking a clausal subject introduced with a complementizer *de* or *que*. Two cases are taken into account in this EPTD:

- complemented infinitives with the polarised features `cpl` ← `de` and `mood` ∼ `inf`,
- complemented finite clauses with the polarised features `cpl` ← `que` and `mood` ∼ `subj`.

In the two cases, the verb provides the subject with a negative feature `cat ← cs`, a positive feature `funct → subj` whereas the negative feature `sent_type ← decl` expresses that the expected subject must be a declarative clause.

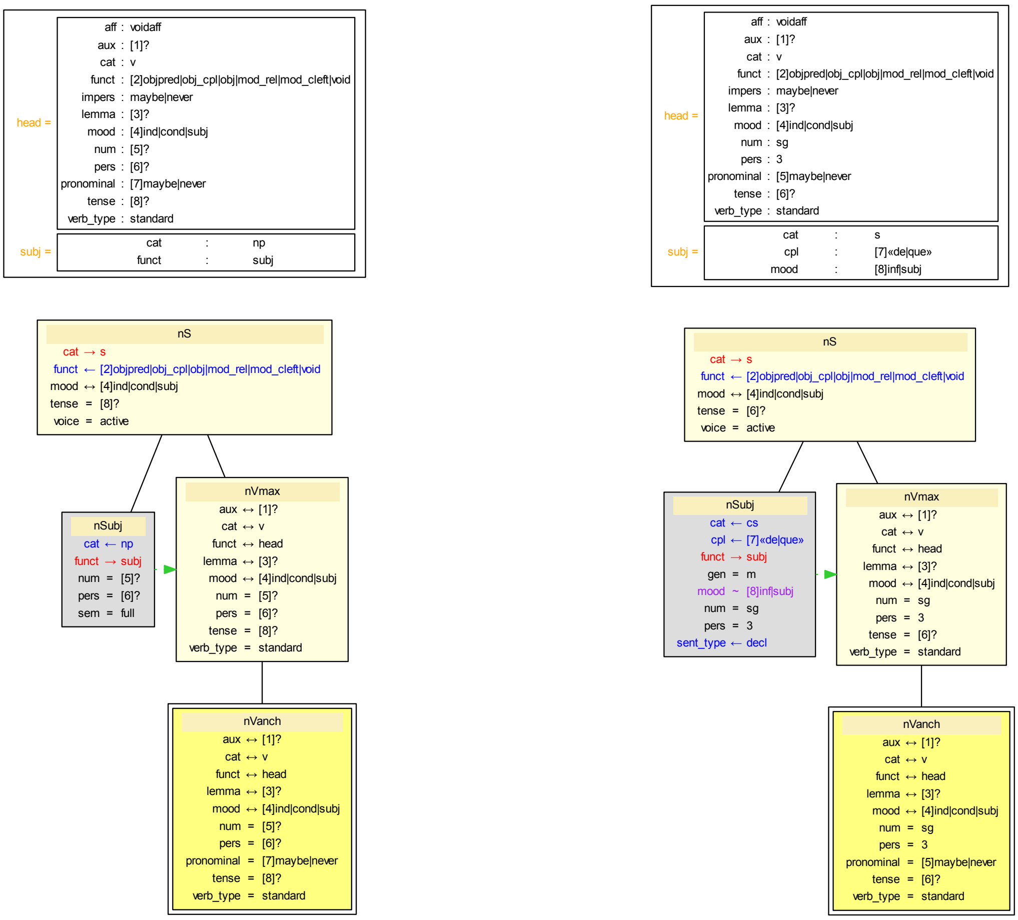

Figure 3.13: EPTDs defined by the NP_VACTIVE and S_VACTIVE classes for verbs in a finite mood.

The two ground classes NP_VACTIVE and S_VACTIVE are combined with complement classes taken from the COMPLEMENT module to build the active diatheses of verbs with complements. We will not give an exhaustive presentation of all combinations. We merely present the example of the NP_VACTIVE_PPINF-SUBJPRED class, resulting from the conjunction of two classes: N_VACTIVE and INDIRECTPREDICATEINFINITIVECLAUSE.

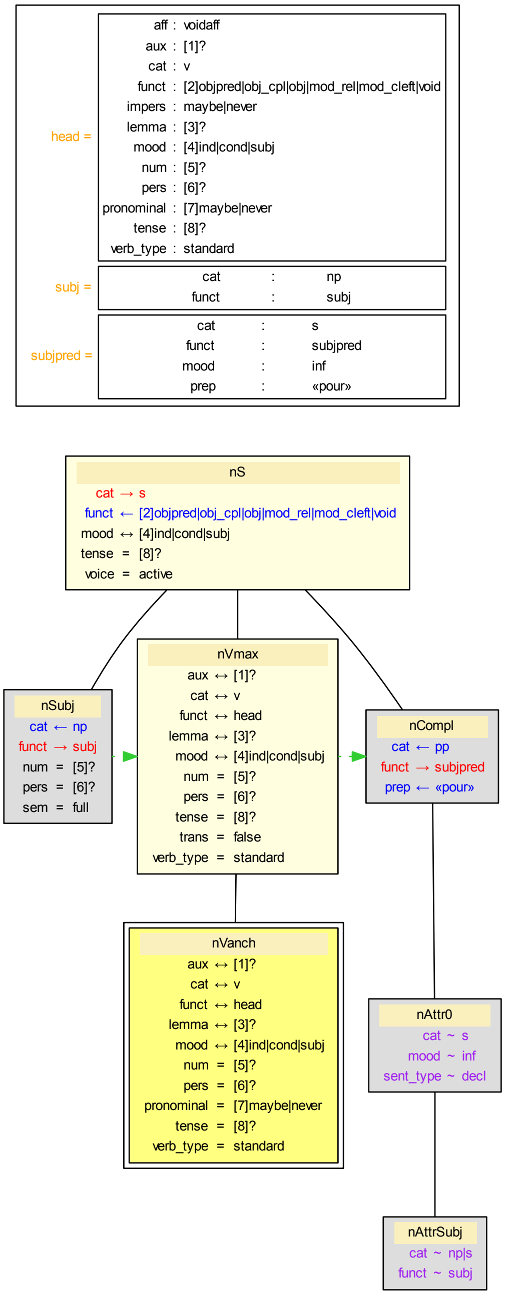

Figure 3.14: EPTD defined by the NP_VACTIVE_PPINF-SUBJPRED class for verbs in finite moods

This class defines different EPTDs, one of which is used by the verb *passe* in the following example.

(3.32) *Marie* **passe** *pour* *être* *une* *femme* *intelligente* *.*
Marie looks being a woman clever .
Marie looks being a clever woman.

This EPTD is shown on Figure 3.14

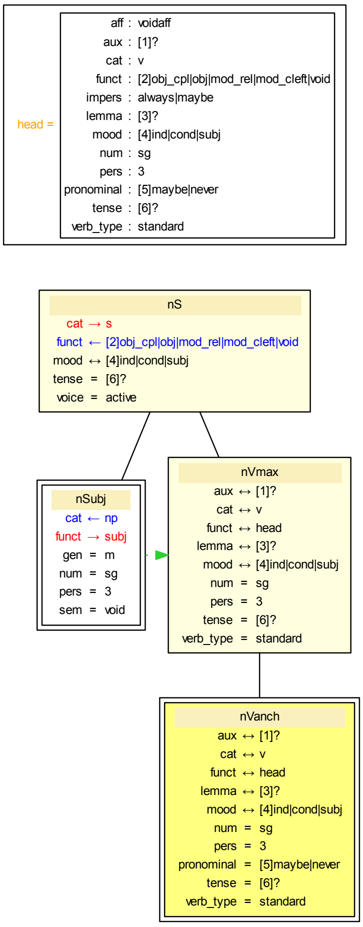

Figure 3.15: EPTD defined by the IL_VACTIVE class for verbs in some finite moods

3.4.2 The verbImpersonalDiathesis module

The VERBIMPERSONALDIATHESIS module includes 24 classes. It concerns the active, passive and middle diatheses of verbs used with an impersonal subject. In most cases, verbs are finite but it may occur that they are infinitive if they are complement of modal verbs, such as in sentence *il peut pleuvoir aujourd'hui (it may be raining today).*

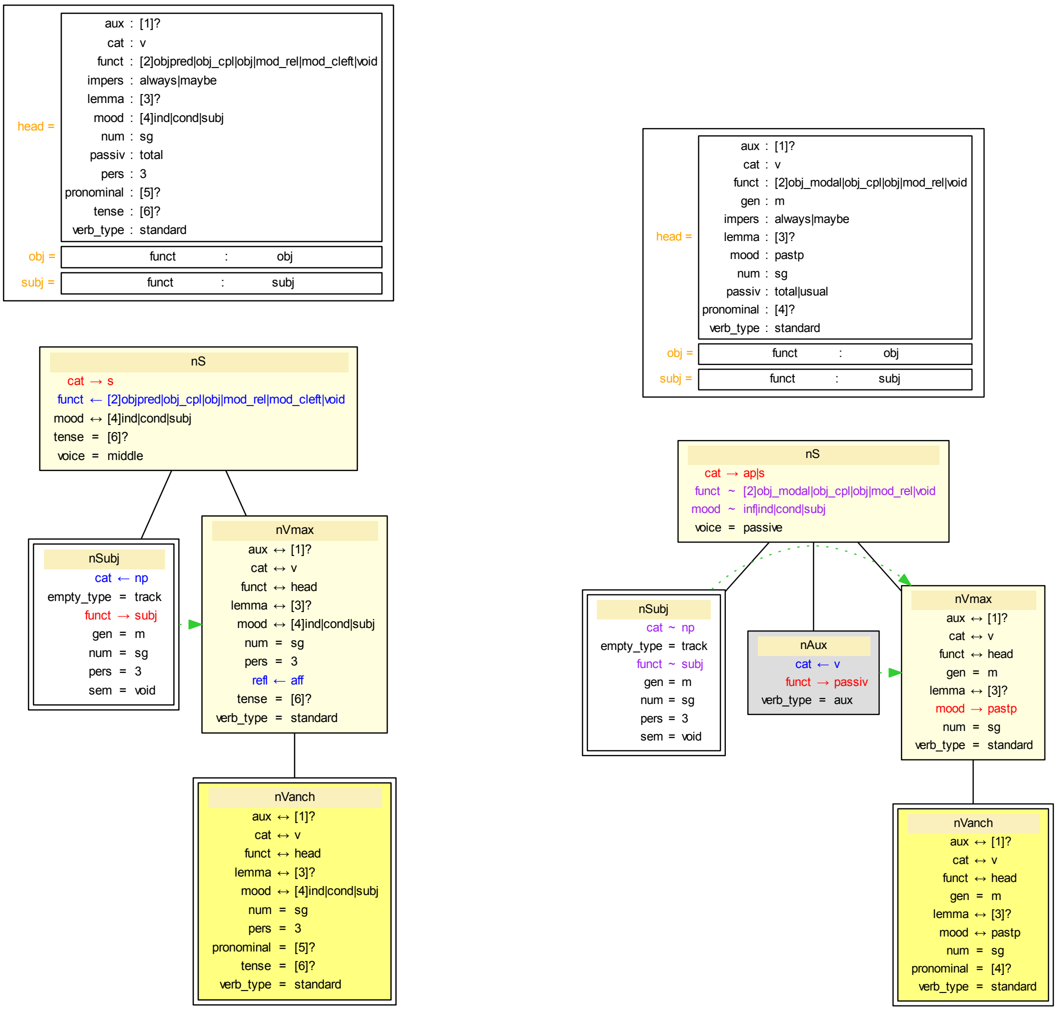

Figure 3.16: EPTDs defined by the Il_Vmiddle and Il_Vpassive classes for verbs in some finite moods

Three ground classes, Il_Vactive, Il_Vmiddle and Il_Vpassive, correspond to the three voices. Figures 3.15 and 3.16 show the EPTDs defined by these classes for some finite moods. They respectively inherit theActiveMorphology, MiddleMorphology and PassiveMorphology classes. In the three EPTDs, the *nSubj* node represents the phonologically empty trace of the impersonal subject *il*, which comes from its status of clitic.

The difference between the Il_Vactive class and the other ones is that the verb can be used in an impersonal construction without any argument, whereas the Il_Vpassive and Il_Vmiddle classes require an argument for the verb, which is the subject in the canonical construction of the verb.This argument is called the *logical subject* of the verb.

Here are examples illustrating the three basic classes.

(3.33) *Il* **pleut** *aujourd'hui .*
it is raining today .
it is raining today.

(3.34) *Il* **arrive** *deux personnes aujourd'hui .*
it is arriving two persons today .
Two persons are arriving today.

(3.35) *Il a été* **vendu** *beaucoup de voitures .*
it has been sold a lot of cars .
it has been sold a lot of cars.

(3.36) *Il se* **vend** *beaucoup de voitures .*
It sells a lot of cars .
A lot of cars are sold.

(3.37) *Il a été* **décidé** *que Jean parte .*
It was decided that Jean go .
It was decided that Jean go.

In Sentences (3.34), (3.35), (3.36) and (3.37), the logical subject is respectively *deux personnes*, *beaucoup de voitures*, *beaucoup de voitures* and *que Jean parte*. So the EPTDs defined by Il_Vpassive and Il_Vmiddle classes cannot be used alone. They must be combined with EPTD coming from the Complement module and giving the different forms of the logical subject.

More generally, complex classes are created by combining the three ground classes with classes coming from the Complement module. Theses classes represent the verbs with their complements in an impersonal construction. The Il_Vpassive_Sinter_agtNP class, for instance, represents the impersonal passive diathesis with an interrogative clause as the logical subject and an agent complement. It results from the conjunction of two classes: Il_Vpassive_Sinter and Agent. The Il_Vpassive_Sinter class itself results from the conjunction of the Il_Vpassive and InterrogativeClauseObject classes.

In the parsing of Example (3.38), the verb *demandé* is used with an EPTD generated by the Il_Vpassive_Sinter_agtNP class. Figure 3.17 shows this EPTD.

(3.38) *Il a été demandé par Jean si Marie venait .*
it has been asked by Jean if Marie was coming .
It has been asked by Jean if Marie was coming.

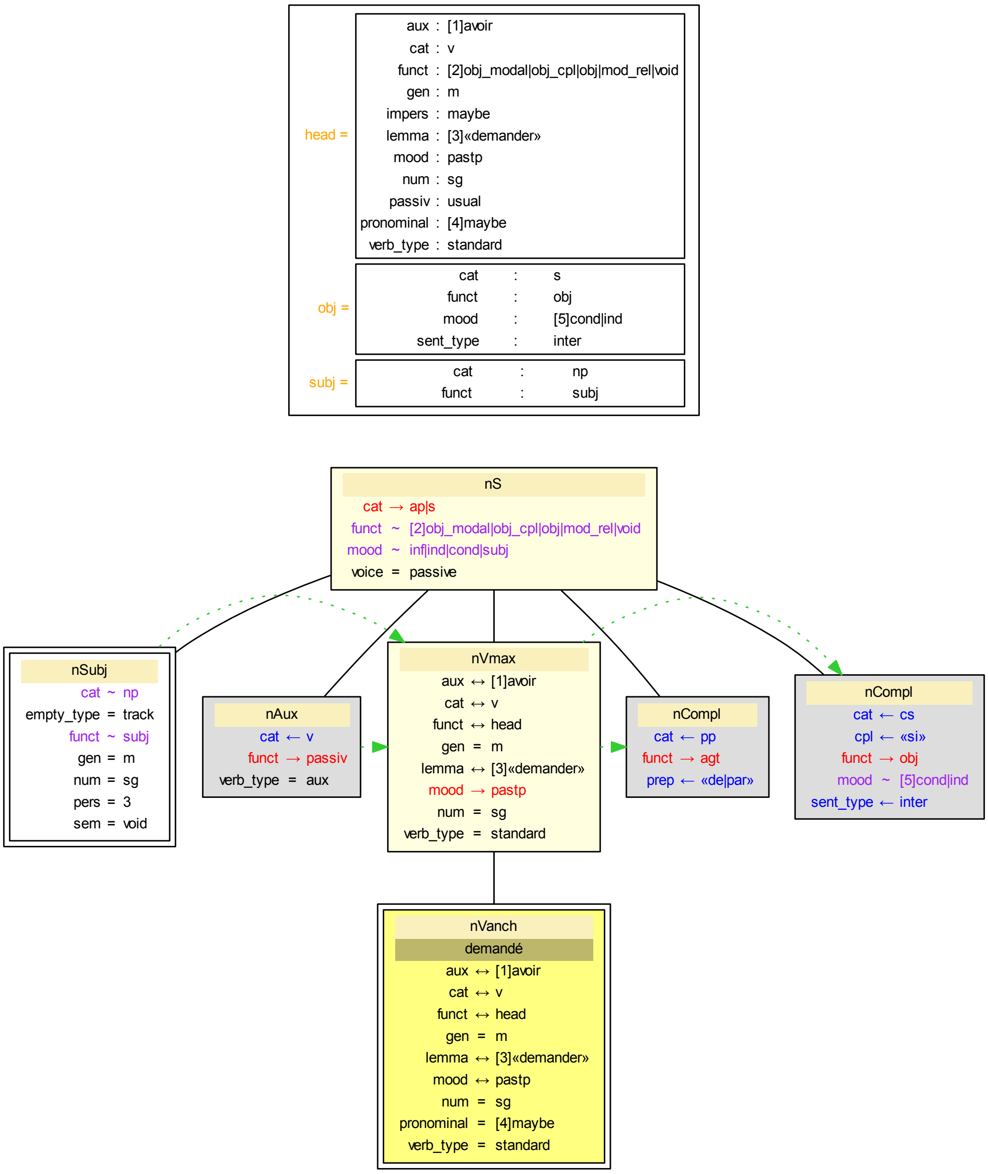

Figure 3.17: EPTD associated with *demandé* in the parsing of *Il a été demandé par Jean si Marie venait.*

3.5 The VERB module of verb families

The VERB module gathers all terminal classes of the verb grammar, except for tense auxiliaries. It includes the different families of standard verbs but also the modal and causative auxiliaries.

3.5.1 The families of standard verbs

For standard verbs, a terminal class represents all diatheses corresponding to a subcategorization frame defining a family. For instance, the N0_V_S1 class corresponds to transitive verbs with a nominal subject and a clausal direct object. It is the disjunction of the following classes representing different diatheses: NP_Vactive_Sinf, NP_Vactive_CSinf, NP_Vactive_quelSfin, NP_Vactive_Sinter, S_Vpassive, S_Vpassive_agtNP, S_Vmiddle, Il_Vpassive_queSfin, Il_Vpassive_Sinter, Il_Vpassive_deSinf, Il_Vmiddle_queSfin, Il_Vpassive_deSinf_agtNP, Il_Vpassive_queSfin_agtNP, Il_Vpassive_Sinter_agtNP. Here are examples illustrating all these different classes taken in the same order.

(3.39) *L'ingénieur* **souhaite** *diriger* *l'entreprise* *.*
the engineer hopes to drive the company .
The engineer hopes to drive the company.

(3.40) *L'ingénieur* **propose** *de* *diriger* *l'entreprise* *.*
the engineer proposes to drive the company .
The engineer proposes to drive the company.

(3.41) *L'ingénieur* **propose** *que* *Marie* *dirige* *l'entreprise* *.*
the engineer proposes that Marie drive the company .
The engineer proposes that Marie drive the company.

(3.42) *Jean* **demande** *quand* *l'ingénieur* *vient* *.*
Jean asks when the engineer is coming .
Jean asks when the engineer is coming.

(3.43) *Que* *Marie* *dirige* *l'entreprise* *est* *fortement* **souhaité** *.*
that Marie drives the-company is strongly hoped .
That Marie drives the company is strongly hoped.

(3.44) *Que* *Marie* *dirige* *l'entreprise* *est* *fortement* **souhaité** *par* *Jean* *.*
that Marie drives the company is strongly hoped by Jean .
That Marie drives the company is strongly hoped by Jean.

(3.45) *Que* *Marie* *aille* *diriger* *l'entreprise* *se* **dit** *en* *ce* *moment*
that Marie is going to drive the company is said in that moment
.
.

That Marie is going to drive the company is said in that moment.

(3.46) *Il est* **souhaité** *que Marie dirige l'entreprise .*
it is hoped that Marie drive the company .
It is hoped that Marie drive the company.

(3.47) *Il a été* **demandé** *pourquoi Jean partait .*
it was asked why Jean was leaving .
It was asked why Jean was leaving.

(3.48) *Il est* **envisagé** *de vendre l'entreprise .*
it is envisaged to sell the company .
It is envisaged to sell the company.

(3.49) *Il se* **dit** *que l'entreprise sera vendue .*
it is said that the company will be sold .
It is said that the company will be sold.

(3.50) *Il est* **envisagé** *par Jean de vendre l'entreprise .*
it is envisaged by Jean to sell the company .
It is envisaged by Jean to sell the company.

(3.51) *Il est* **souhaité** *par Jean que Marie dirige l'entreprise .*
it is hoped by Jean that Marie drive the company .
It is hoped by Jean that Marie drive the company.

(3.52) *Il a été* **demandé** *par le directeur pourquoi Jean partait .*
it was asked by the director why Jean was leaving .
It was asked by the director why Jean was leaving.

When a family represents verbs with two or three complements, the corresponding class is created by inheritance of a class corresponding to verbs with one complement and by conjunction of classes corresponding to additional complements. For instance, the NP0_V_NP1_PP2_PP3 class results from the conjunction of the NP0_V_NP1_PP2 and the NominalIndirectObject classes. The NP0_V_NP1_PP2 class itself results form the conjunction of the NP0_V_NP1 and the NominalIndirectObject classes.

In the following Example (3.53), the verb *transporter* is associated with an EPTD generated by the NP0_V_NP1_PP2_PP3 class. Figure 3.18 shows this EPTD.

(3.53) *Marie fait transporter un colis de Paris à Marseille .*
Marie makes carry a parcel from Paris to Marseille .
Marie makes a parcel carry from Paris to Marseille.

head =	aff	:	voidaff
	aux	:	[1]avoir
	cat	:	v
	funct	:	[2]subjpred|subj|objpred|obj_prep|obj_modal|obj_cpl|obj|mod_rel|iobj|caus|app|void
	impers	:	maybe
	lemma	:	[3]«transporter»
	mood	:	inf
	pronominal	:	[4]maybe
	trans	:	true
	verb_type	:	standard
iobj1 =	cat	:	np
	funct	:	iobj
	prep	:	[5]«deloc»
iobj2 =	cat	:	np
	funct	:	iobj
	prep	:	[6]«loc»
obj =	cat	:	np
	funct	:	obj
subj =	cat	:	np
	funct	:	subj

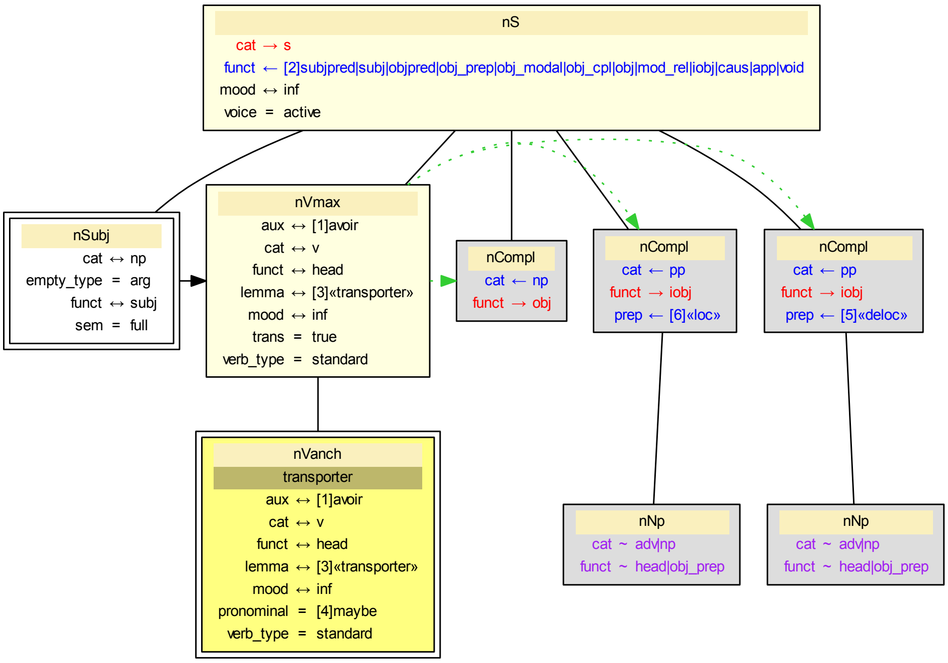

Figure 3.18: EPTD associated with *transporter* used in the parsing of sentence *Marie fait transporter un colis de Paris à Marseille*

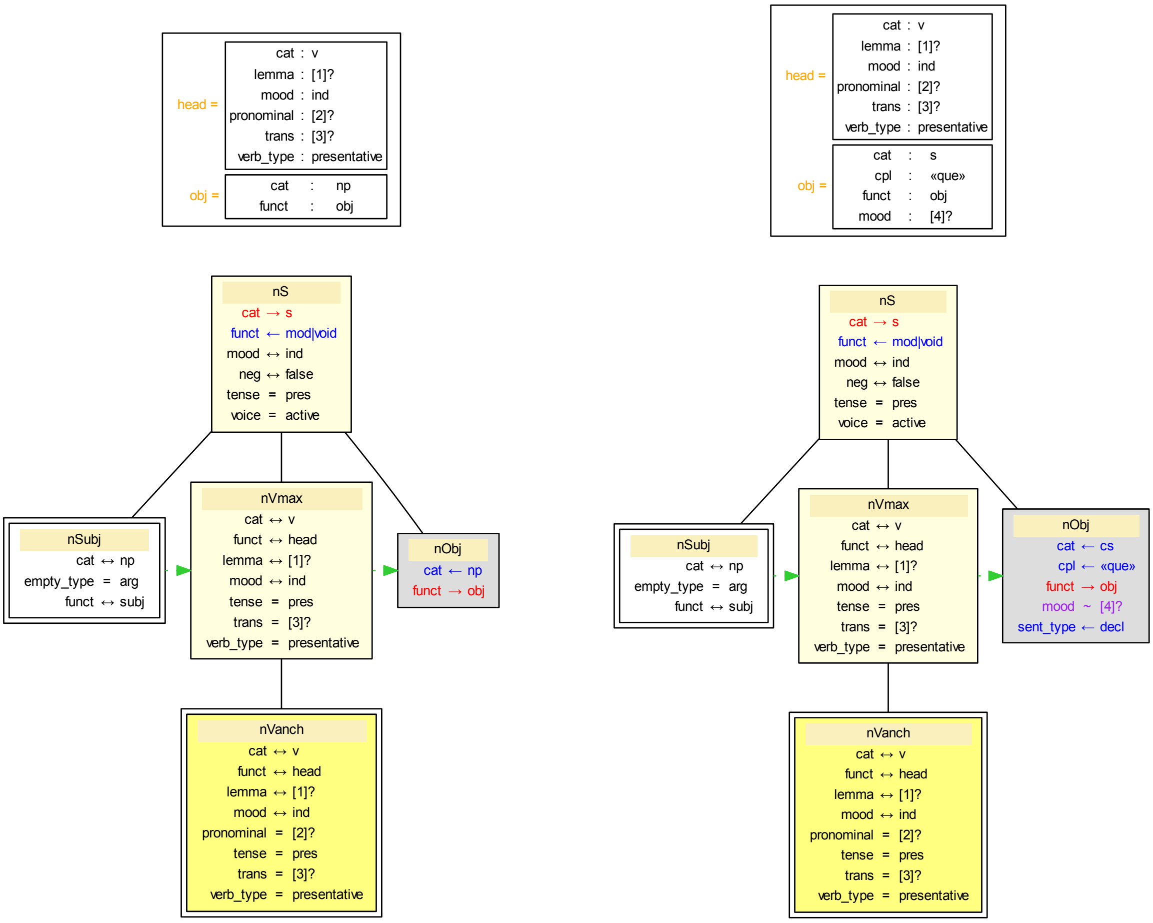

Figure 3.19: EPTDs defined by the VPRESENT_C1 class

3.5.2 Presentatives

Presentatives, like *voici* and *voilà*, are usually considered as adverbs. In FRIGRAM, they are considered as special verbs and their EPTDs are defined by the VPRESENT_C1 and VPRESENT_C1_C2OBJPRED classes. The first class concerns presentatives with a direct object, which may be nominal or clausal. The second class add an object predicate complement. Here are examples illustrating these different cases.

(3.54) *Le* *livre* *que* **voici** *est* *facile* *à* *lire* .
The book that you see here is easy to read .
The following book is easy to read.

(3.55) **Voici** *que* *Jean* *vient* *.*
Here that Jean is coming .

Jean is coming.

(3.56) *La* **voilà** *tranquille* *.*
Her there quiet .

She is quiet now.

(3.57) *Le* **voici** *qui* *arrive* *.*
Him here who is coming .

He is coming now.

The two first sentences illustrate the Vpresent_C1 class and the two last ones the Vpresent_C1_C2objpred class. Figure 3.19 shows the two EPTDs defined by the Vpresent_C1 class. The difference between the two EPTDs lies in the nature of the object: nominal or clausal.

3.5.3 Modal verbs

Modal verbs are dealt with in a particular way because they are transparent with respect to their subject, which is constrained by their object infinitive. The following examples illustrate this property.

(3.58) *Il* ***commence*** *à* *pleuvoir* *.*
It is beginning to be raining .

It is beginning to be raining.

(3.59) *Jean* ***peut*** *venir* *.*
Jean may come .

Jean may come.

(3.60) *Travailler* *la nuit* ***doit*** *être* *difficile* *.*
To work at night shall be difficult .

To work at night shall be difficult

A specific class Vmodal_C1inf defines the EPTDs of modal verbs from the conjunction of two classes: ActiveMorphology and PredComplement. The transparency of the modal verb with respect to its subject is expressed with co-references between its subject *nSubj* and the subject *nComplSubj* of the infinitive that is its direct object.

There is a first alternative related to the form of this subject: nominal, non complemented infinitive or complemented clause. Then there is another alternative related to the form of the infinitive object depending on whether it is introduced by a complementizer or not.

Figure 3.20 shows an instantiation of the Vmodal_C1inf for the verb *doit* used in Sentence (3.60).

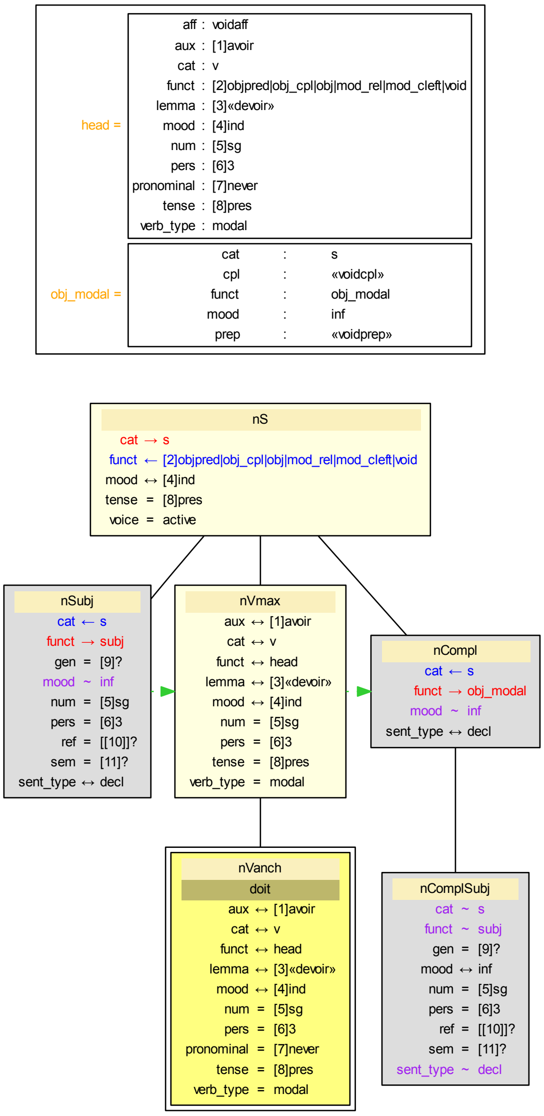

Figure 3.20: EPTD associated with *doit* used in the parsing of sentence *travailler la nuit doit être difficile*

3.5.4 Causative verbs

In a causative construction, a causative auxiliary (*faire* or *laisser* in French) combines with an infinitive in the active voice. Here are examples illustrating this construction. For every sentence, the causative auxiliary and the complement infinitive are in bold.

(3.61) *Jean* *a* ***fait*** ***venir*** *Marie* *aujourd'hui* *.*
Jean got to come Marie today .

Jean got Marie to come today.

(3.62) *Manger* *beaucoup* ***fait*** ***dormir*** *.*
To-eat a-lot causes sleeping .

To eat a lot causes sleeping.

(3.63) *Que* *Marie* *mange* *beaucoup* *la* ***fait*** ***dormir*** *.*
That Marie eats a-lot her makes sleeping .

That Marie eats a lot makes her sleeping.

(3.64) *Jean* *s'est* ***fait*** ***contrôler*** *.*
Jean himself has made control .

Jean has made control himself.

(3.65) *Jean* ***fait*** ***balayer*** *la* *cour* *par* *Marie* *.*
Jean asks to sweep the yard by Marie .

Jean asks Marie to sweep the yard.

(3.66) *Jean* ***fait*** ***balayer*** *la* *cour* *à* *Marie* *.*
Jean asks to sweep the yard to Marie .

Jean asks Marie to sweep the yard.

(3.67) *Jean* ***fait*** *se* ***rencontrer*** *les* *ingénieurs* *aujourd'hui* *.*
Jean makes meet the engineers today .

Jean makes the engineers meet today.

In FRIGRAM, causative auxiliaries are considered as special full verbs and the associated ground class is CausativeVerb. An alternative way of modelling them would be to consider them as actual auxiliaries, like tense or passive auxiliaries. It would require to add a specific entry in the grammar for all infinitives likely to take a causative auxiliary, which would increase the size of the grammar and the lexical ambiguity in parsing. Another drawback, which will be explained later, is related to the addition of a specific direct object aroused by the causative construction.

The CausativeVerb class inherits the ActiveMorphology class and it adds particular features to node *nSubj* according to the form of the subject: nominal or clausal.

Figure 3.21 shows the two EPTDs defined by this class for verbs in the conditional, indicative or subjunctive mood. The left EPTD corresponds to a nominal subject and the right one to a clausal subject. The infinitive object of the causative auxiliary is represented with node *nCaus* and its void subject with node *nCausSubj*.

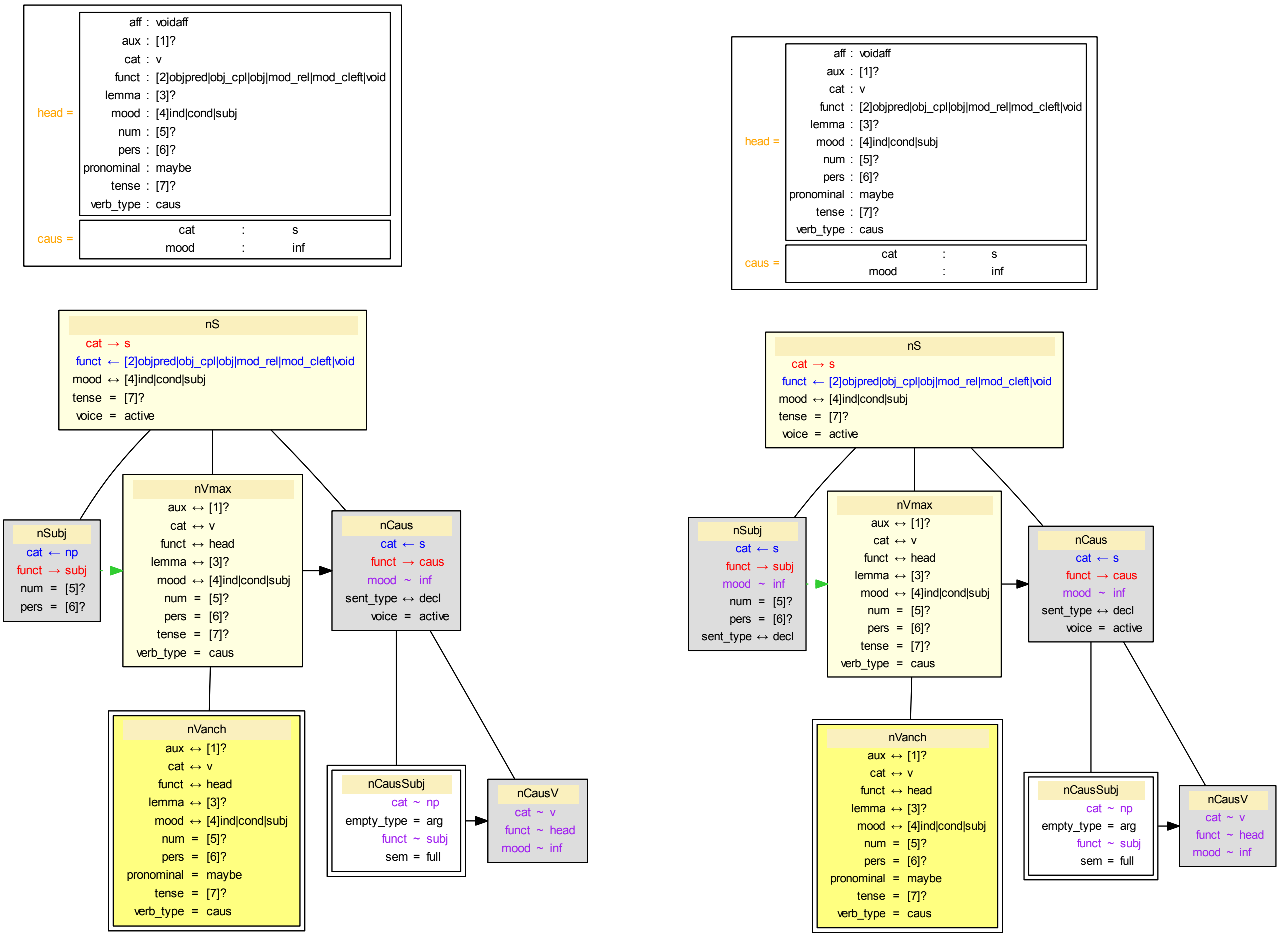

Figure 3.21: EPTDs defined by the CausativeVerb class

The causative construction entails specific complements:

- The patient that is caused to perform the action expressed by the infinitive is the object of the causative verb if the verb is intransitive. Sentences (3.61) and (3.63) illustrate this construction and the CausativeVerbWithObj, inheriting the CausativeVerb class, represents causative auxiliaries used in this construction.

- The patient that is caused to perform the action expressed by the infinitive is an indirect complement of the causative auxiliary with a dative or agent function if the infinitive is transitive. Sentences (3.65) and (3.66) illustrate this construction and the CausativeVerbWithAObj and CausativeVerbWithAgt classes, inheriting the CausativeVerb class, represent causative verbs used in the same construction.

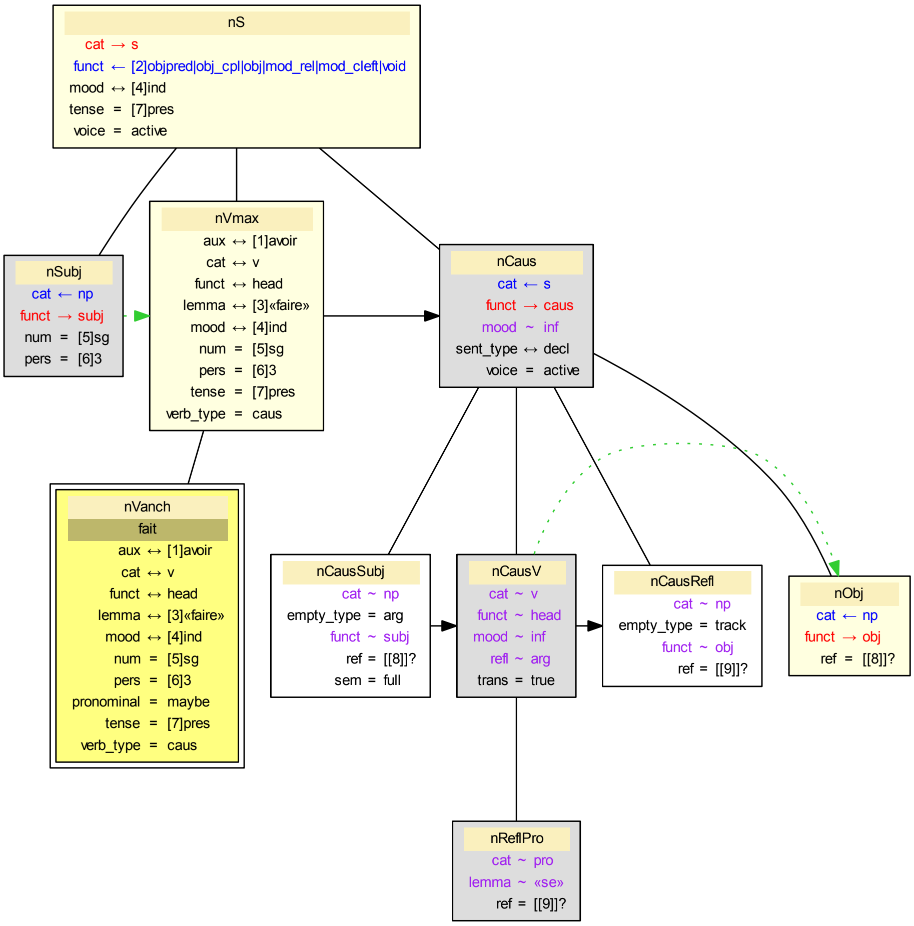

Figure 3.22: EPTD defined by the CausativeVerbWithObj class

These specific complements could be put as complements of the causative verb in parallel with the infinitive headed by the caused verb. A major drawback of this representation is that it does not allow the infinitive to be interrupted by such a complement, which occurs sometimes in French, as the the example *Jean fait demander par Marie un médecin (Jean demands a doctor by Marie).*

To avoid this problem, the additional complements introduced by the causative verb are put in the infinitive clause as complements of the caused verb. A drawback of the representation is that the caused verb can take two direct objects, which occurs in Example (3.67): the first object is the reflexive pronoun *se* and the second object is the additional complement *les ingénieurs* introduced by the causative verb *fait*.

Figure 3.22 shows the EPTD used for the causative verb in this example. Node *nCausRefl* represents a trace of the object reflexive pronoun *se* and node *nObj* represents the additional object *les ingénieurs* introduced by the causative verb *fait*.

When the complements of the caused verb are clitic pronouns, they rise to the causative verb if they are reflexive pronoun referring to the subject of the causative verb, or if they are non reflexive. This will be developed in subsection 6.2.3 of chapter 6.

Chapter 4

Nouns

The NOUN module gathers all classes anchored with common and proper nouns.

4.1 Interfaces with the lexicon

Nouns are characterised in interfaces with the feature `head.cat = n`. Their morphological features and some syntactic properties are gathered in the `head` feature:

- `det_type`: its value indicates the type of determiner that can combine with the noun to build a noun phrase; the possible values are: `de` (*de* determiner), `def` (definite), `dem` (demonstrative), `indef` (indefinite), `neg` (negative), `num` (numeral), `part` (partitive), `poss` (possessive), `voiddet` (non determiner);
- `funct`: it gives the possible functions of the noun, which are `app` (apposition), `obj_cpl`(object of a complementizer in clauses with ellipsis), `mod` (modifier), `obj` (direct object), `objpred` (object predicate), `obj_prep` (object of a preposition), `subj` (subject), `subjpred` (subject predicate);
- `gen`: it indicates the gender of the noun with the values `f` and `m`;
- `noun_type`: it gives a subcategorization of nouns according to semantic properties; it respectively takes the values `abstr` (abstract), `anim` (animate), `count` (inanimate countable noun), `mass` (inanimate mass noun), `propnoun` (proper noun);
- `num`: it gives the number of the noun: `pl` (plural) or `sg` (singular);
- `sent_type`: when the noun is the head of a nominal sentence, it gives the type of the sentence, `excl` (exclamative) or `imper` (imperative).

The properties of special classes of nouns are described with features which are put in parallel with the head feature and describe the required syntactic arguments or the governor of the noun:

- `gov`: the governor of the noun when it behaves as a modifier,

- `iobj1`: first required complement,
- `iobj2`: second required complement,
- `verb`: the support verb that has the noun as its direct object.

4.2 Common and proper nouns

Two basic classes CommonNoun and ProperNoun define the skeleton of all classes related to common and proper nouns. They produce the PTDs of Figure 4.1.

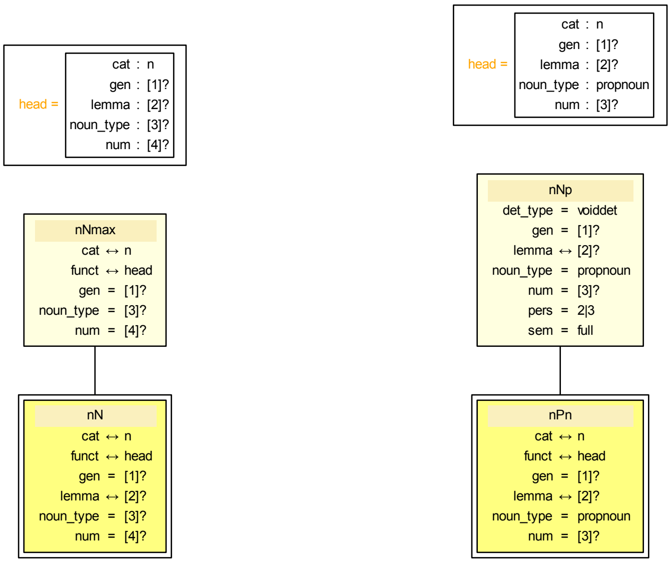

Figure 4.1: The PTD defined by the CommonNoun and ProperNoun classes

All features of the PTD defined by the CommonNoun class are neutral or saturated. The PTD will be completed in a sub-class with a mother node for *nNmax*, and *nNmax* will be the head of this new node.

In the PTD defined by the ProperNoun class, the feature `det_type = voiddet` expresses that a proper noun requires no determiner.

Proper nouns have two possible functions illustrated with the following examples.

(4.1) ***Marie*** *est interrogée par Jean .*
Marie is asked by Jean .
Marie is asked by Jean.

(4.2) ***Madame*** *la directrice arrive* .
Madam the director is coming .
Madam the director is coming.

(4.3) *Mon ami* **Jean Martin** *est ingénieur* .
My friend Jean Martin is engineer .
My friend Jean Martin is engineer.

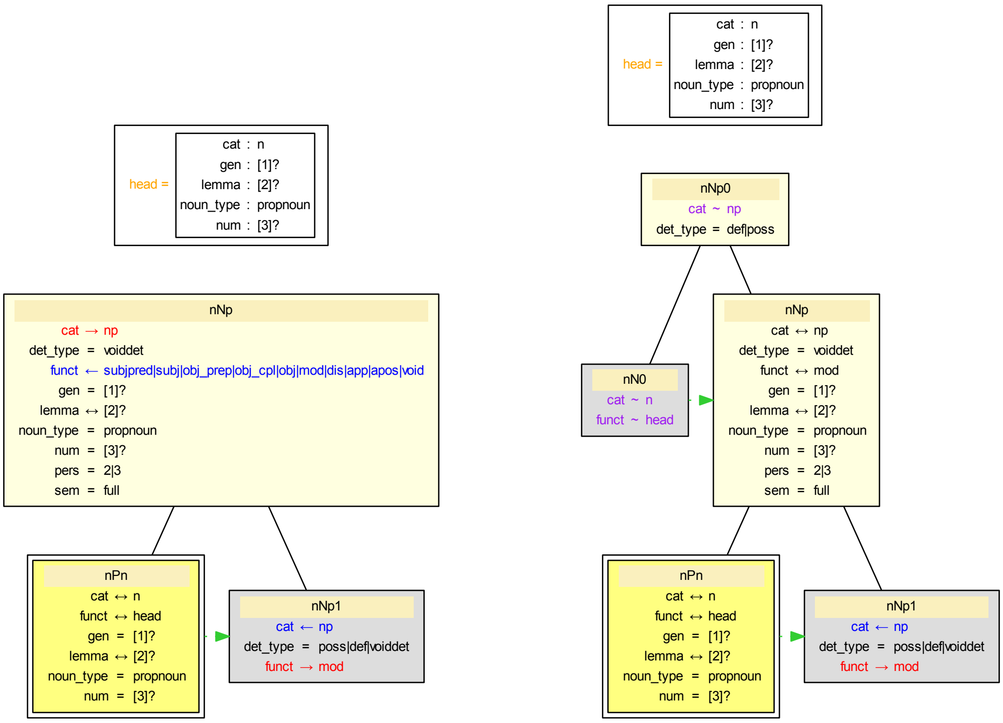

Figure 4.2: EPTDs defined by the Nproper and Nproper_NP1gov classes

The two functions are expressed with two classes inheriting the ProperNoun class:

- The Nproper class expresses the usual function of noun phrases. It defines two EPTDs. The first one, illustrated with Sentence (4.1), is a simple copy of the PTD defined by the ProperNoun class. The second one, illustrated with Sentence (4.2), adds the need of modifying the proper noun with any noun phrase. It is shown on the left of Figure 4.2. In our example, the proper noun *Madame* is modified by the noun phrase *la directrice*.
- the Nproper_NP1gov class expresses that proper nouns can be modifiers of any noun phrases, which is illustrated with Sentence (4.2). As the previous class, it

defines two EPTDs, according to the fact that the proper noun is or is not modified by a noun phrase. In our example, *Jean* illustrates the second case, because it modifies *Mon ami* and it is modified by *Martin*. The EPTD used in this last case is shown on the right of Figure 4.2.

4.3 The syntactic functions of common nouns

The following examples illustrate various syntactic functions of common nouns (the concerned common nouns are in bold).

(4.4) *La* **fille** *est grande .*
The girl is tall .
The girl is tall.

(4.5) *La réunion a* **lieu** *demain .*
The meeting takes place tomorrow .
The meeting takes place tomorrow.

(4.6) *Jean est* **ingénieur** *.*
Jean is engineer .
Jean is engineer.

(4.7) *Jean travaille toute la* **nuit** *.*
Jean works all the night .
Jean works all the night.

(4.8) **Dommage** *que Jean ne soit pas venu !*
Pity that Jean did not come !
It is a pity that Jean did not come!

(4.9) *Jean a lu un roman* **fleuve** *.*
Jean has read a novel-fleuve .
Jean has read a novel-fleuve.

Common nouns can be heads of noun phrases, as Sentences (4.4) and (4.5) show it, which is expressed with the NounPhraseHead class. This class defines the PTD of Figure 4.3. In this PTD, the positive feature `cat` → `np` and the negative feature `funct` express the fact that a common noun can be the head of any noun phrase, which can receive various functions in a sentence.

The NounPhraseHead class is specialised in two subclasses:

- DetCommonNoun that expects a determiner to build a noun phrase, as Sentence (4.4) shows it,
- ObjectCommonNounWithSupportVerb that combine with a support verb to build an idiomatic expression, as Sentence (4.5) shows it.

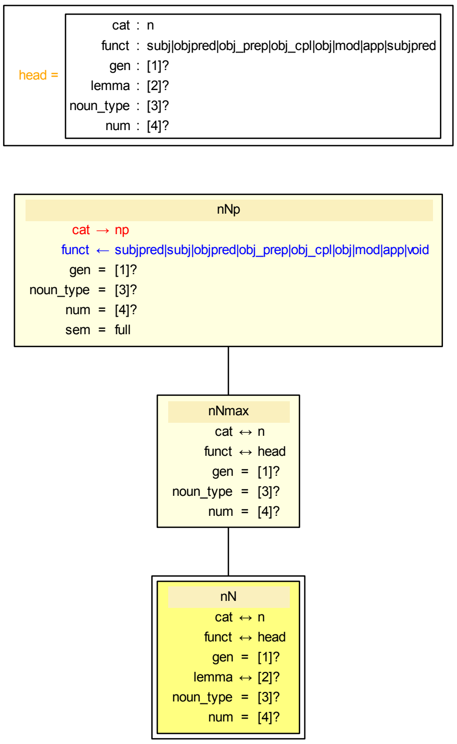

Figure 4.3: The PTD defined by the NounPhraseHead class

Figure 4.4 shows the two EPTDs defined by these classes. In the left EPTD, a node *nDet* represents the expected determiner and its ability to interact with a real determiner is expressed with the negative feature `cat` ← `det` and the positive feature `funct` → `det`. In the right EPTD, node *nNp* represents the maximal projection of the anchored common noun. Its positive feature `cat` → `np` and its negative feature `funct` ← `obj` express the fact that the common noun will be the object of the support verb, the kernel of which is represented with *nVmax*. For constructions with support verbs, there are some arguments to consider the support verb as the head of the sentence and other arguments to consider that the predicative noun is the head. We have chosen the verb as the head of the sentence represented with *nS*.

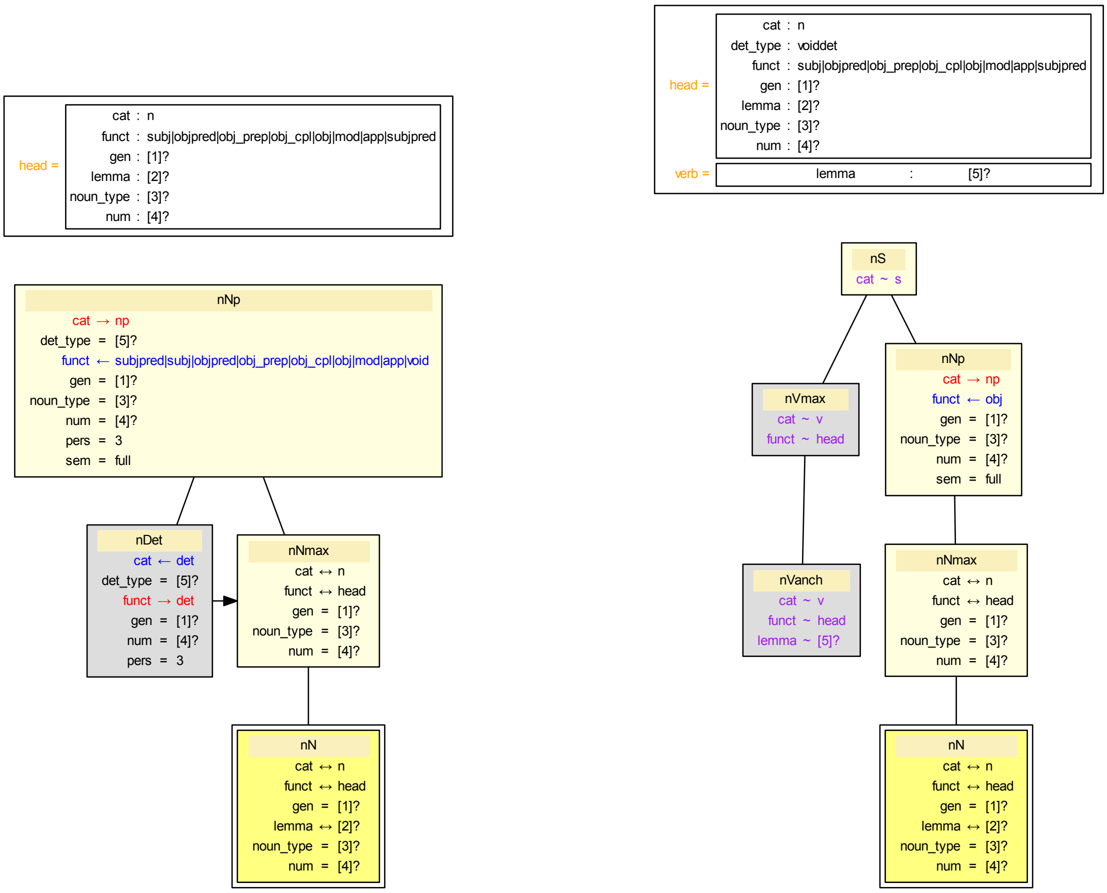

Figure 4.4: The EPTDs defined by the DetCommonNoun and ObjectCommonNounWithSupportVerb classes

Since the lemma of the support verb is determined by the noun, there is a feature `verb.lemma` which is shared by the interface and by the node *nVanch* of the PTD. For Example (4.5), in the EPTD anchored with *lieu*, the feature takes the value *avoir*.

A common noun can be used as a predicate complement, as Sentence (4.6) shows it. A particular class PredicativeCommonNoun, inheriting the CommonNoun class, defines a corresponding EPTD shown in Figure 4.5. A common noun playing the role of a predicate complement behaves as an adjective, hence the maximal projection of the anchored common noun is represented with a node *nAp* with the positive feature `cat` → `ap`. Like for any adjectival phrase, an empty node *nSubjAp* represents the subject of the adjectival phrase.

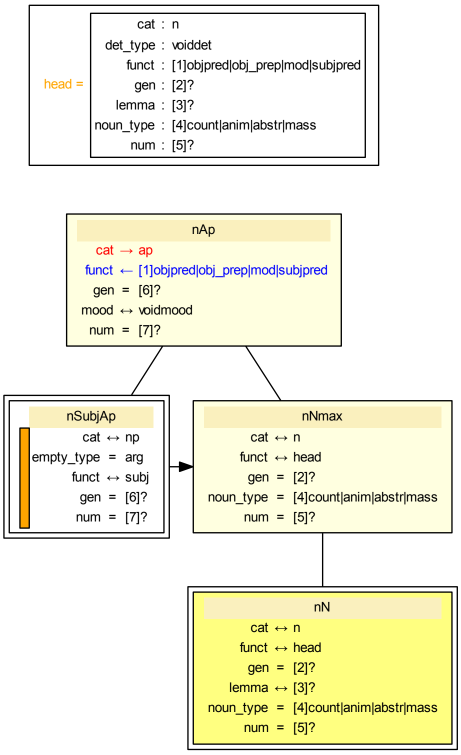

Figure 4.5: The EPTD defined by the PredicativeCommonNoun class

Some commons nouns can be used as temporal complements without preposition, as Sentence (4.7) shows it. The TemporalNoun class expresses this use defining the EPTD of Figure 4.6. This EPTD expresses that the anchored common noun is the head of a noun phrase represented with node *nNp* and acting as a circumstantial complement in a sentence or a noun phrase represented with node *nC0*.

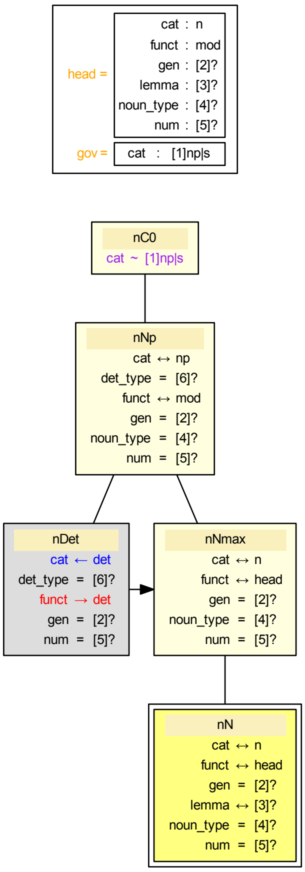

Figure 4.6: The EPTD defined by the TemporalNoun class

Other particular common nouns can be heads of sentences, as Example (4.8) shows it. The SentenceHeadNoun class expresses this use defining the left EPTD of Figure 4.7.

Some common nouns are used in an attributive function like adjectives, as Sentence (4.9) illustrates it. It is modelled with the Nattr class and shown on the right of Figure 4.7. Most often, the modifier nouns strongly constrain the noun they modify, which is expressed with a feature `lemma`, which is shared by the interface and by the node *nN0* of the PTD.

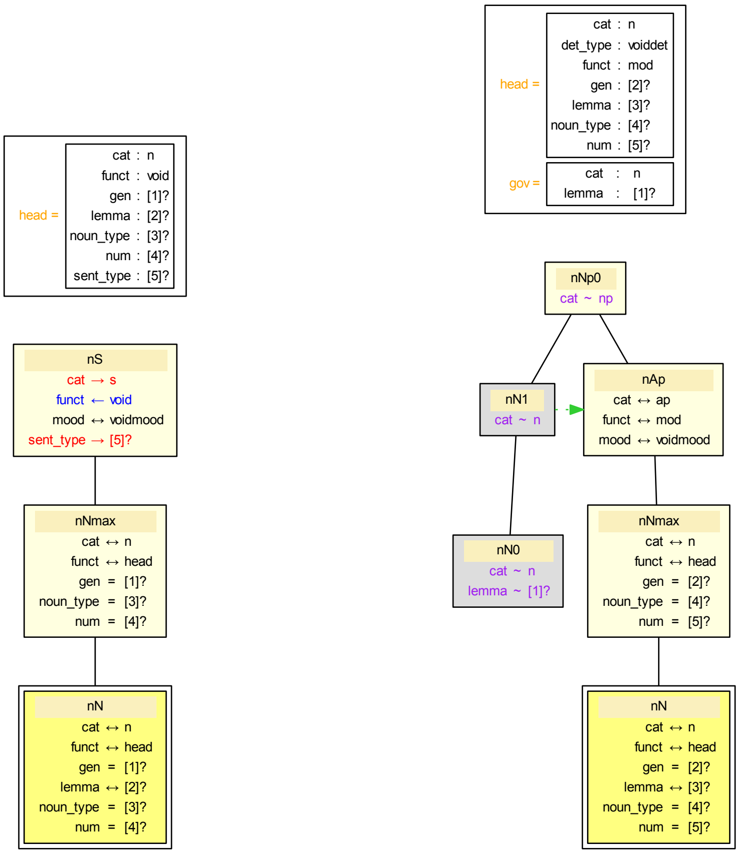

Figure 4.7: The EPTDs defined by the SENTENCEHEADNOUN and NATTR classes

4.4 Nouns with required complements

Some common nouns required various kinds of complements as the following examples illustrate it. In the examples, the concerned nouns are in bold.

(4.10) *Jean a pris* **contact** *avec l' entreprise* .
Jean has made contact with the company .
Jean has made contact with the company.

(4.11) *J'ai l'* **accord** *pour que Jean vienne* .
I have the agreement for that Jean comes .
I have the agreement for Jean coming.

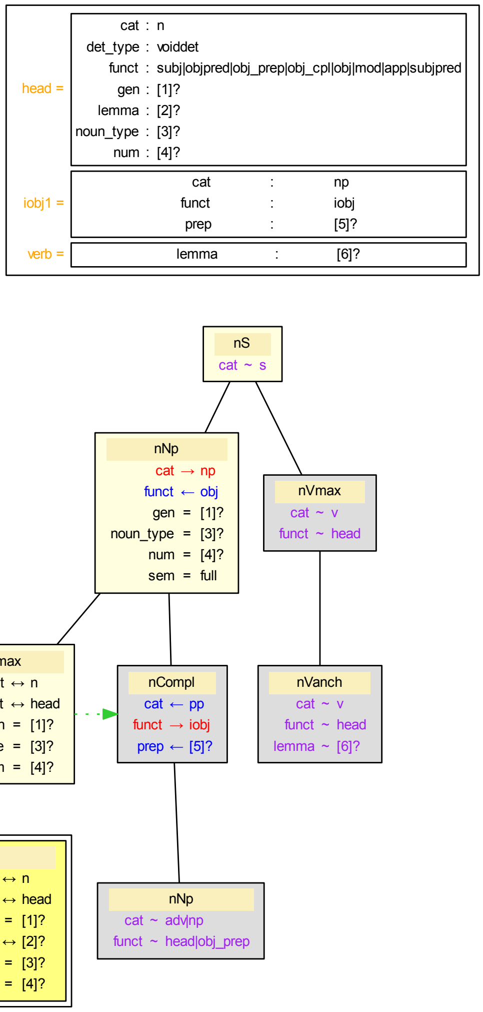

Figure 4.8: An EPTD defined by the N_PP1nom class

(4.12) *La* **crainte** *de* *venir* *est* *grande* .
The fear of coming is great .
The fear of coming is great.

(4.13) **Dommage** *que* *Jean* *vienne* *seulement* *demain* *!*
Pity that Jean come only tomorrow !
It is a pity that Jean come only tomorrow!

(4.14) *L'* **invitation** *de* *l'* *entreprise* *à* *l'ingénieur* *est* *arrivée* .
The invitation of the company to the engineer has come .

The invitation of the company to the engineer has come.

(4.15) *La* **crainte** *de Jean de venir est grande .*
The fear of Jean of coming is great .
The Jean's fear of coming is great.

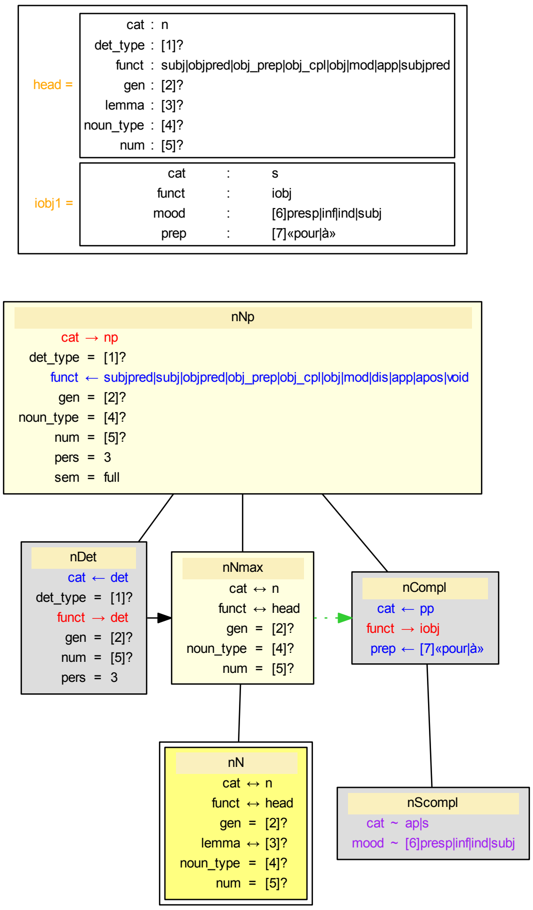

Figure 4.9: An EPTD defined by the N_PP1SENT class

As the examples above show it, all kinds of complements of a common noun can combine with all syntactic functions of this nouns; we assume an exception: when the common noun is an attribute of another common noun. Hence, as a preliminary of the definition of the related classes, a class N is the disjunction of the DetCommonNoun, PredicativeCommonNoun, ObjectCommonNounWithSupportVerb, TemporalNoun and SENTENCEHEADNOUN classes. The N class defines 7 EPTDs because the TEMPORALNOUN and SENTENCEHEADNOUN classes each one correspond to two EPTDs.

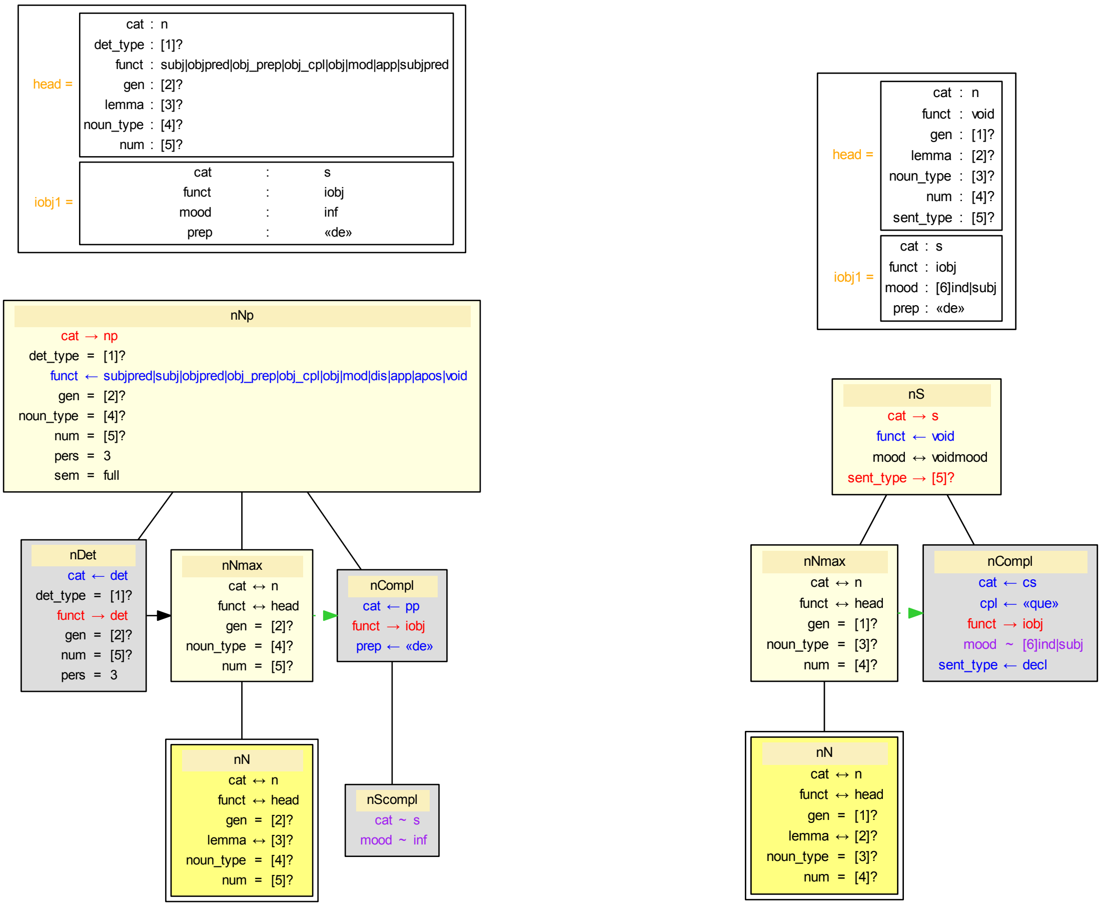

Figure 4.10: Two EPTDs defined by the NDES1 class

Then, the N class combines with the classes of the COMPLEMENT module to produce EPTDs of nouns requiring complements.

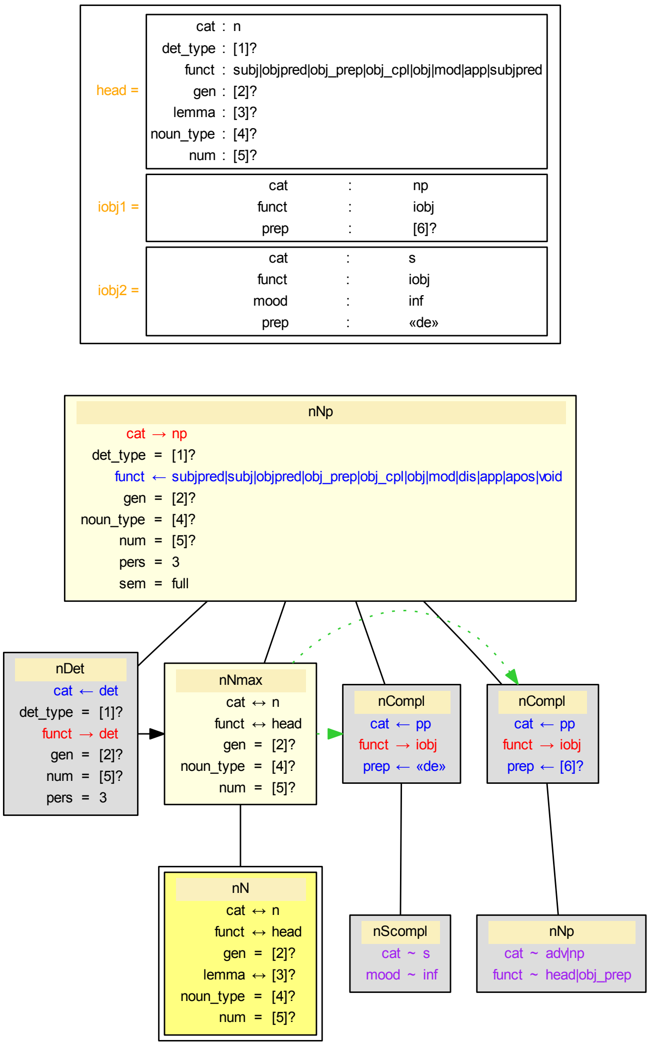

Figure 4.11: An EPTD defined by the N_PP1nom_deS2 class

For instance, by conjunction of the N class and the NominalIndirectObject class, we obtains the a class N_PP1nom, which models the syntactic construction of any common noun requiring a nominal complement introduced by a preposition. This class defines 12 EPTDs because for every syntactic function of a common noun, there are two possible nominal complements: a common noun without determiner or a complete noun phrase. Figure 4.8 shows the EPTD corresponding to Example (4.10).

Another class N_PP1sent models the syntactic construction of any common noun requiring a clausal complement, except if the complement is introduced with preposition *de*, which is a particular case. It defines 6 EPTDs corresponding to the 6 EPTDs for the different syntactic functions of common nouns without complements. Figure 4.9 shows the EPTD corresponding to Example (4.11).

Commons nouns requiring clausal complements introduced with preposition *de* are defined with a specific class N_deS1 because for the complement there is a systematic alternation between an infinitive introduced with *de* and a finite clause introduced with *que*. Examples (4.12) and (4.13) express this alternation and Figure 4.10 show the EPTDs used to parse the examples.

Like a verb, a common noun can require several complements, which is expressed by the N_PP1nom_PP2nom, N_PP1nom_PP2sent and N_PP1nom_deS2 classes and illustrated with the Examples (4.14) and (4.15).

Figure 4.11 presents the EPTD used to parse Sentence (4.15) and defined by the N_PP1_deS2 class.

Chapter 5

Determiners

The characteristic feature of determiners is that they are required by common nouns to build noun phrases.

5.1 Interfaces with the lexicon

Determiners are characterised in interfaces with the feature `head.cat = det`. Their morphological features and some syntactic properties are gathered in the `head` feature:

- `det_type`: it gives the type of the determiner, `de` (the unique determiner *de*), `def` (definite), `dem` (demonstrative), `indef` (indefinite), `neg` (negative), `num` (numeral), `part` (partitive), `poss` (possessive), `super` (superlative) and `voiddet` (no determiner); we have more kinds of determiners than the grammarians have defined because we have split the class of indefinites into 5 sub-classes: `num` and `part`, which select countable and mass nouns, `de` for the unique determiner *de*, which has a specific behaviour[1], `neg` for negative determiners and `indef` for standard indefinites;
- `gen`: it indicates the gender of the determiner with the values `f` and `m`;
- `num`: it gives the number of the determiner, `pl` (plural) or `sg` (singular);

5.2 Standard determiners

The basic class Determiner, which is shared by all determiners, is very general. It defines the PTD shown on Figure 5.1. In this PTD, node *nNp* represents the noun phrase that has the common noun *nN* as its head and is determined by the anchor determiner. The positive feature `cat` → `det` and the negative feature `funct` ← `det` labelling the maximal projection *nDetmax* of the determiner, express the one-to-one possible interaction with a common noun requiring a determiner.

[1] Since in the XMG language, it is not possible to express that an indefinite determiner is different from *de*, we use this trick to express the difference.

The DET_N1 class defines the EPTD of standard determiners: definite, indefinite (in the large sense), demonstrative and possessive determiners. With respect to the Determiner class, it only adds the value `def|dem|indef|part|poss` to feature `det_type` in the interface.

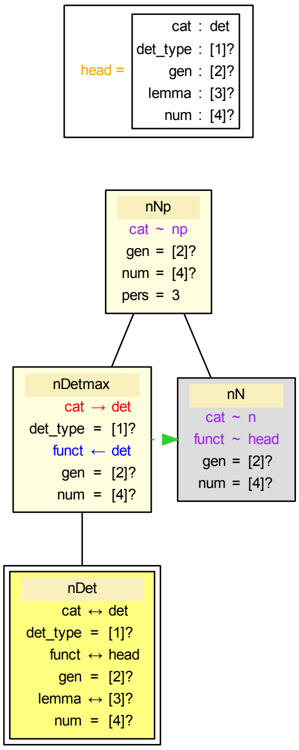

Figure 5.1: PTD defined by the Determiner class

5.3 Related Determiners

There are specific determiners that depend on other words in their environment. There are two kinds of determiners having this property: negative determiners and the indefinite determiner *de*.

5.3.1 Negative determiners

Negative determiners, like *aucun*, are paired with the clitic *ne* put before the verb head of the clause receiving the noun phrase introduced by the determiner, but the position of the noun phrase is relatively free inside the clause, as the following examples show it.

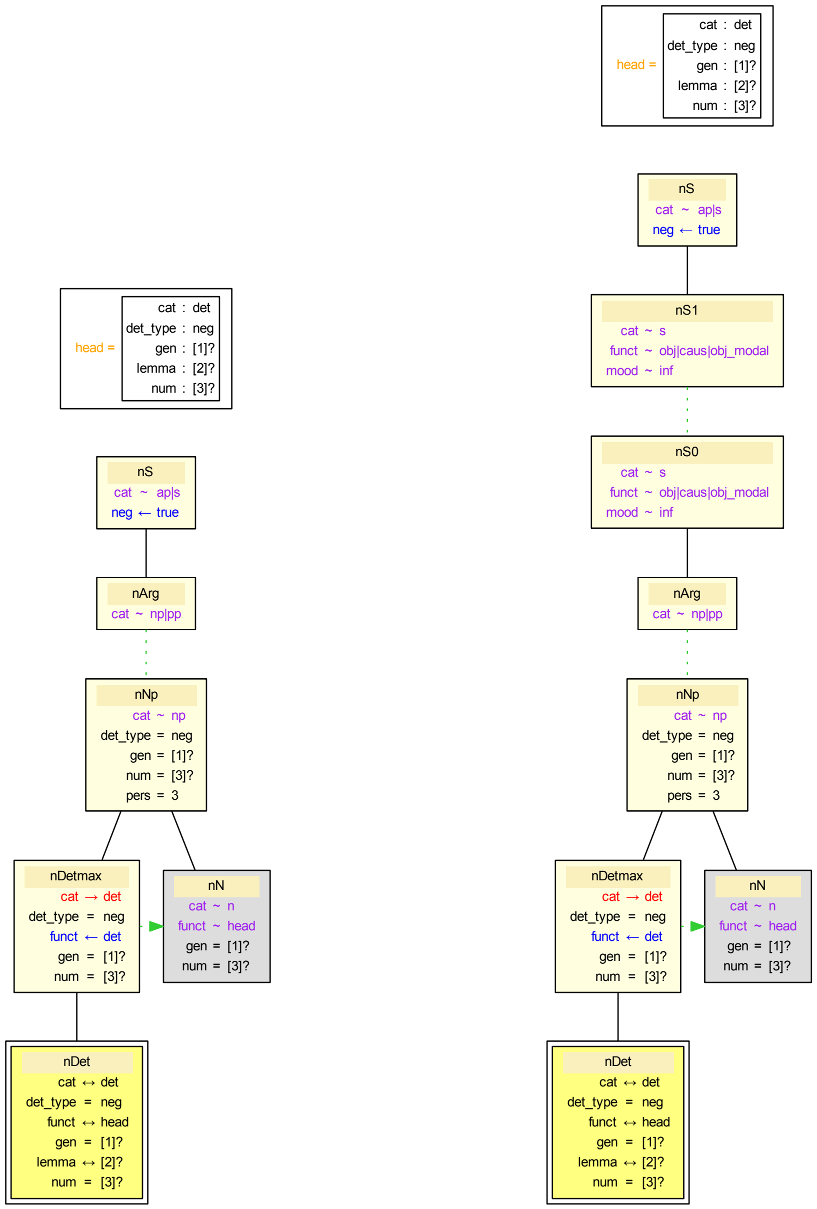

Figure 5.2: EPTDs defined by the DETneg_N1 class

(5.1)	*Marie*	*ne*	*connaît*	*la*	*femme*	*d'*	*aucun*	*ingénieur*	*.*
	Marie	does not	know	the	wife	of	any	engineer	.

Marie knows no wife of any engineer.

(5.2) *Marie ne pense connaître la femme d' aucun ingénieur .*
Marie does not think to know the wife of any engineer .
Marie think to knows the wife of no engineer.

The DETneg class defines the EPTDs associated with any negative determiner, which is shown in Figure 5.2. There are two EPTDs because there are two cases:

- the particle *ne* is in the same clause as the negative determiner, which is illustrated with Sentence (5.1);
- the particle *ne* is in a clause embedding the clause including the negative determiner, which is illustrated with Sentence (5.2).

In both cases, node *nNp* represents the noun phrase determined by the negative determiner anchored at node *nDet*. In both examples, the noun phrase is *aucun ingénieur*. It can be embedded more or less deeply in a prepositional or noun phrase, which is represented by an underspecified dominance relation from node *nArg* over node *nNp*. Node *nArg* corresponds to the noun phrase *la femme d'aucun ingénieur*.

In both cases, node *nS* represents the clause aimed at receive the particle *ne*, which is expressed with the negative feature `neg` ← `true`. The dual feature `neg` → `true` will be provided by the particle *ne*.

The difference between the two cases lies in the relation between node *nArg* and node *nS*. In the left EPTD, node *nArg* is a direct daughter of *nS*. The right EPTD expresses the possibility for node *nArg* to be embedded in a pileup of infinitives depending on the head verb of the main clause *nS* with a modal, causative or object relation. The most external infinitive is represented with node *nS1* and the most internal infinitive is represented with node *nS0*. Between them, there is an underspecified dominance relation. In Sentence (5.2), there is only one infinitive, *connaître la femme d'aucun ingénieur*, for which node *nS1* is merged with node *nS0*.

5.3.2 The indefinite determiner *de*

The indefinite determiner *de* is paired with a negation or with an adjective preceding the noun that it introduces. The examples below illustrate theses cases.

(5.3) *Jean connaît de grandes entreprises .*
Jean knows big companies .
Jean knows big companies.

(5.4) *Jean ne connaît pas d'entreprise .*
Jean does not know company .
Jean knows no company.

(5.5) *Jean ne pense pas connaître d'entreprise .*
Jean does not think to know company .
Jean does not think to know any company.

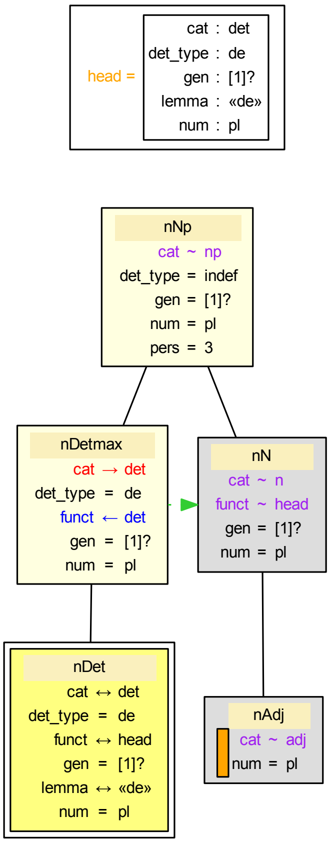

Figure 5.3: EPTD defined by the DETDEADJ_N1 class

According to the two cases, there are two respective classes: DETDEADJ_N1 and DETDENEG_N1. Figure 5.3 shows the EPTD defined by the DETDEADJ_N1 class which is used to parse Sentence (5.3) below. On the figure, node *nAdj* represents an adjective thet must modify the common noun *nN* to built the noun phrase *nNp* with the determiner *nDetmax*.

The DETDENEG_N1 class defines two EPTDs shown on Figure 5.4. They are similar to those shown on Figure 5.2. The only difference lies in restrictions put in the use of node *nNp*. It represents a direct object of the verb head of the sentence including it. Examples (5.4) and (5.5) illustrate the use of the two EPTDs.

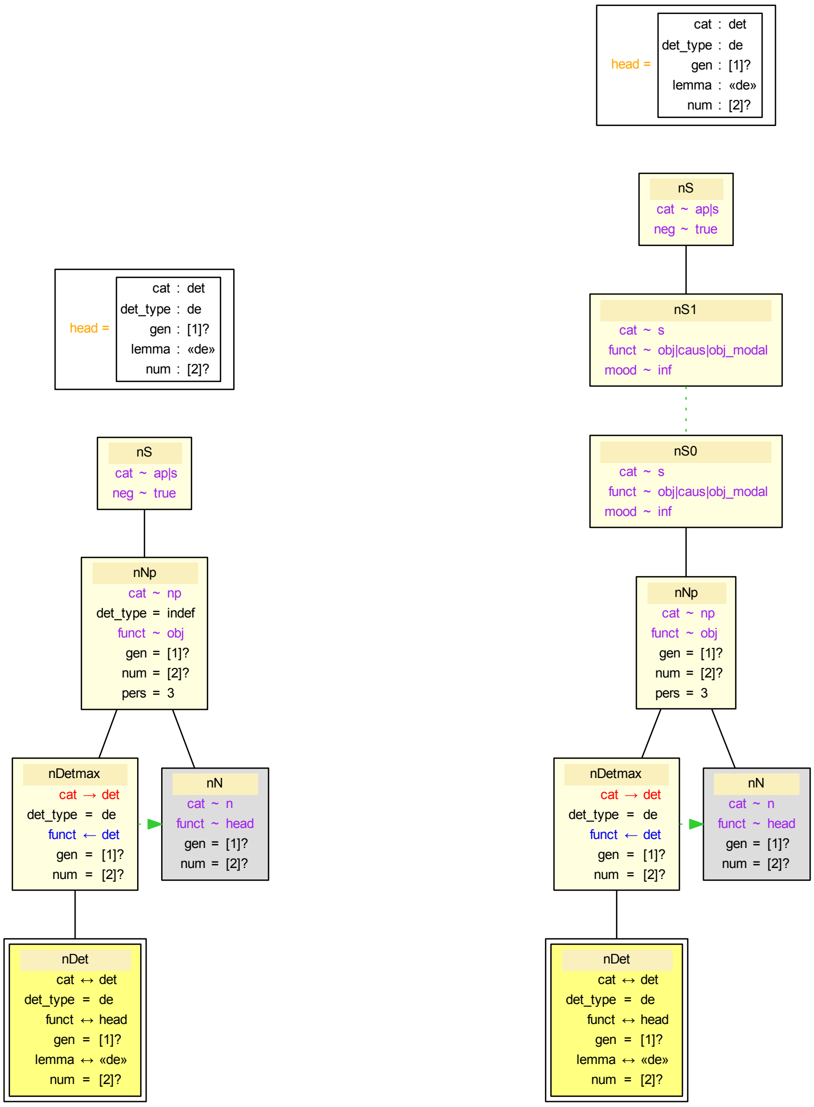

Figure 5.4: EPTDs defined by the DETDENEG_N1 class

Chapter 6

Pronouns

The classes related to pronouns are distributed between two modules: ProClitic for clitic pronouns and Pronoun for disjunctive pronouns. Clitic pronouns behave as quasi affixes of verbs.

6.1 Interfaces with the lexicon

Pronouns are characterised in the interface with the feature `head.cat = pro`. They are ranged in different classes according to the value of the `head.pro_type` feature:

- `clit`: clitic pronouns (*il, le, lui*...),
- `def`: disjunctive definite (personal) pronouns (*lui, lui-même*...),
- `dem`: demonstrative pronouns (*ceci, ça*...),
- `indef`: indefinite pronouns (*chacun, tout*...),
- `inter`: interrogative pronouns (*lequel, que, quel*...),
- `neg`: negative pronouns (*aucun, rien, personne*...),
- `poss`: possessive pronouns (*le sien*...),
- `rel`: relative pronouns (*dont, lequel, que*...).

Other features are used to describe pronouns:

- `aff`: when the pronoun is an affix, except a reflexive affix, it gives the value of this affix, `en`, `le` or `y`;
- `det_type`: it gives the type of the determination corresponding to the pronoun; its values are the same as for the equivalent feature for determiners (see section 5.1);
- `funct`: some clitic pronouns have a syntactic function marked with this feature; the possible functions are `iobj`, `mod`, `obj`, `objpred`, `subj`, `subjpred`, `void`;

- `gen`: it indicates the gender of the pronoun with the values `f` and `m`;
- `num`: it gives the number of the pronoun, `pl` (plural) or `sg` (singular);
- `pers`: it gives the person of the pronoun, `1`, `2` or `3`;
- `sem`: this feature indicates if a subject clitic pronoun has a semantic counterpart with the values `full` and `empty`.

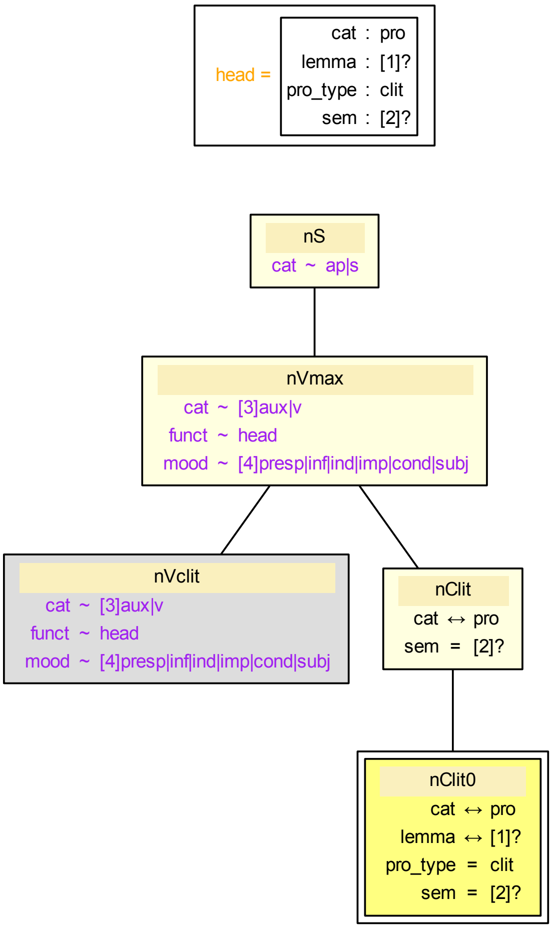

Figure 6.1: PTD defined by the Clitic class

6.2 Clitic pronouns

6.2.1 Affix versus argument clitics

A clitic pronoun is placed side by side near a verb to modify it. The general skeleton of all its EPTDs is defined by the Clitic class and shown on Figure 6.1. A clitic pronoun anchored at node *nClit0* has its maximal projection represented by node *nClit*. The modified verb is represented by node *nVclit* and its maximal projection by node *nVmax*. This one is an immediate sub-constituent of node *nS* representing a clause or an adjectival phrase.

A clitic pronoun can represent a verb or noun argument. It can also be an affix without any argument function. The following examples illustrate these different functions.

(6.1) *Jean* **la** *voit .*
Jean her sees .
Jean sees her.

(6.2) *Jean* **en** *connaît la couleur .*
Jean of it knows the color .
Jean knows the decision of him.

(6.3) *Finis*-**en** *avec ce travail .*
Finish with this work .
Put an end to this work.

(6.4) *Les ordinateurs* **se** *vendent bien .*
The computers themselves sell well .
The computers sell well.

Sentences (6.3) and (6.4) illustrate the two cases of non argument clitics. The first one concerns the clitics *en, y, le*, which are used to modify the semantics of the verb to which they are linked. The corresponding PTD is defined with the AffixClitic class and shown on the left of Figure 6.2. Node *nClit* representing the maximal projection of the clitic carries three polarised features `aff` → `le|en|y`, `cat` → `pro` and `funct` ← `aff`, which will be neutralised by features brought by the EPTD associated with an appropriate verb.

The second case concerns the reflexive clitic *se* which is used to build pronominal verbs or to express the middle voice of some transitive verbs. Sentence (6.4) illustrates middle voice. The EmptyReflexiveClitic class defines these two uses of the *se* clitic. It generates the PTD presented on the right of Figure 6.2. They are two differences with respect to the PTD defined by the AffixClitic class. First, the PTD carries only one polarized feature, the feature `refl` → `aff`, which is attached at the maximal projection of the verb, represented with node *nVmax*. This feature will be neutralised by a dual feature brought by an EPTD associated with the middle voice of a verb or with the active voice of a pronominal verb. Second, there is an agreement in person (`pers`) and number (`num`) between the clitic pronoun represented with node *nClit* and the subject of the verb represented with node *nSubj*.

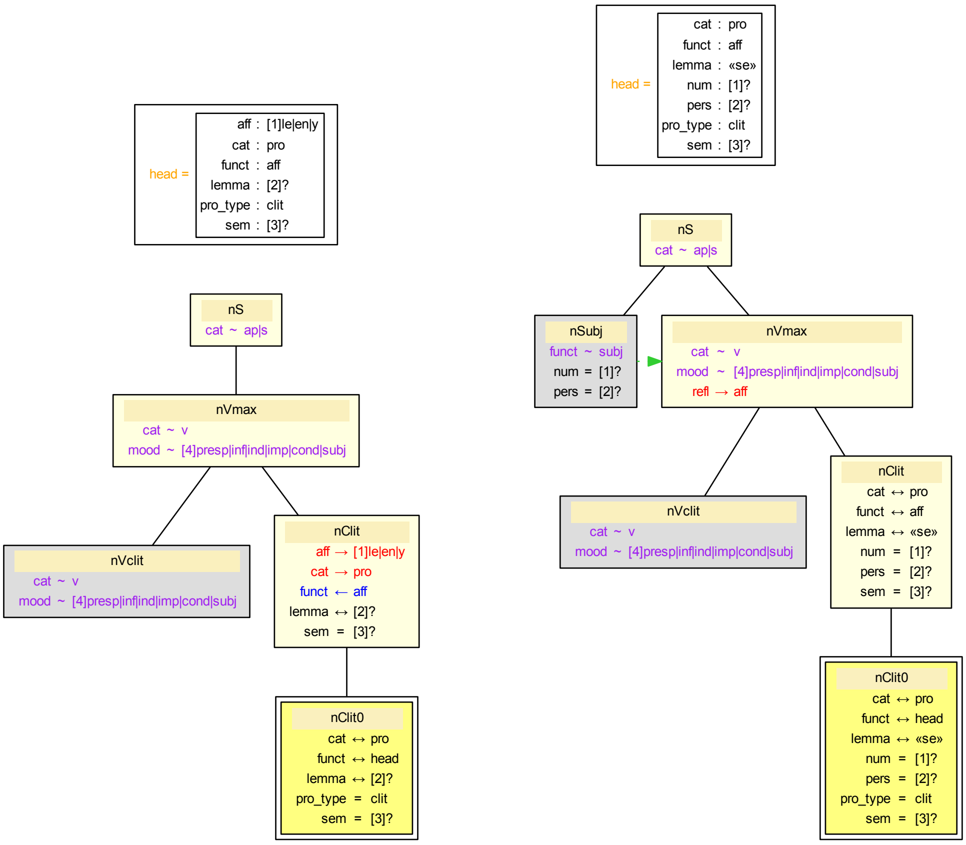

Figure 6.2: PTDs defined by the AffixClitic and EmptyReflexiveClitic classes

In Sentence (6.1), the clitic *la* represents the object of the verb *voit*. In Sentence (6.2), the clitic *en* represents a complement of the noun *couleur*. The use of a clitic pronoun as an argument is defined by the FullConstituentClitic class. This class generates three PTDs according to the function of the clitic, direct or indirect, and to its morphology, variable or invariable. Figure 6.3 shows the PTD defined for clitics that are direct arguments (*il, on, ce, le, se*). The argument is represented with node *nConst*, which is empty or full. This node co-refers to node *nClit*, representing the clitic pronoun. There is no structural relation between *nConst* and the main description, because the relation is variable according to the function of the clitic.

Figure 6.4 shows the PTD defined for clitics that are indirect complement with a variable morphology (*lui, se*). The difference with respect to the previous PTD, is the trace has a complex structure. It is constituted of a prepositional phrase, represented with node *nConst*, which has an empty noun phrase *nNp* as its head. This noun phrase corefers with node *nClit*.

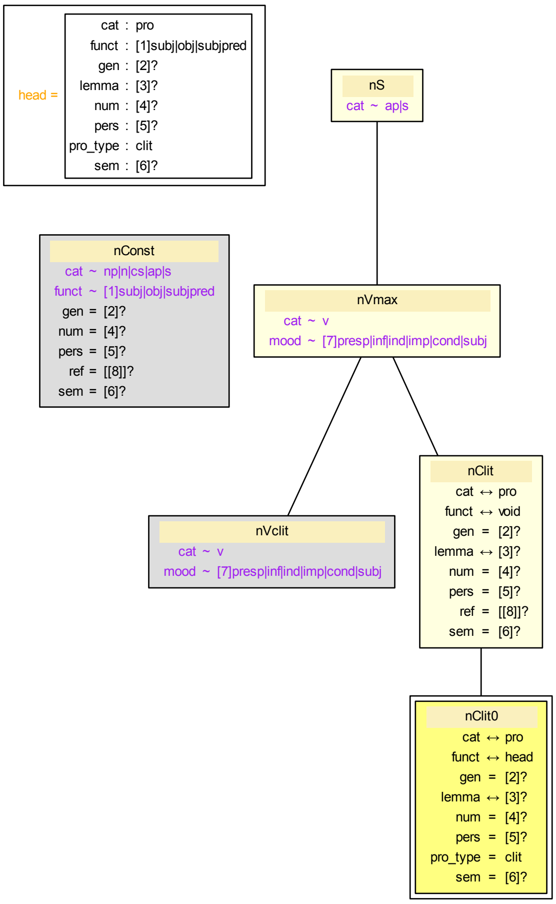

Figure 6.3: PTD defined by the FullConstituentClitic class for clitics that are direct arguments

The FullConstituentClitic class generates a last PTD for indirect and invariable complements (*y*, *en*). The only difference with the previous PTD is that agreement features are absent.

6.2.2 Subject clitic pronouns

A first possible function of clitics is subject. Here are examples with different uses of subject clitics.

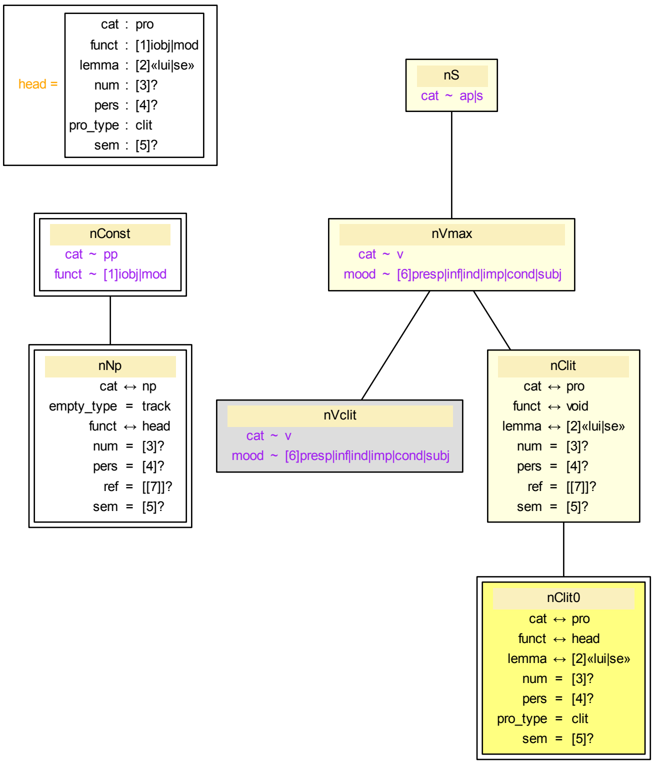

Figure 6.4: PTD defined by the FullConstituentClitic class for clitics that are indirect inflected complements

(6.5) **Il** *pleut* .
It rains .
It rains.

(6.6) *Vient*-**il** *aujourd'hui* ?
Comes he today ?
Does he come today?

(6.7) *Jean* *vient*-**il** *aujourd'hui* ?
Jean comes he today ?
Does Jean come today?

(6.8) *Est*-**ce** *le* *fils* *de* *Jean* ?
Is it the son of Jean ?
Is it Jean's son?

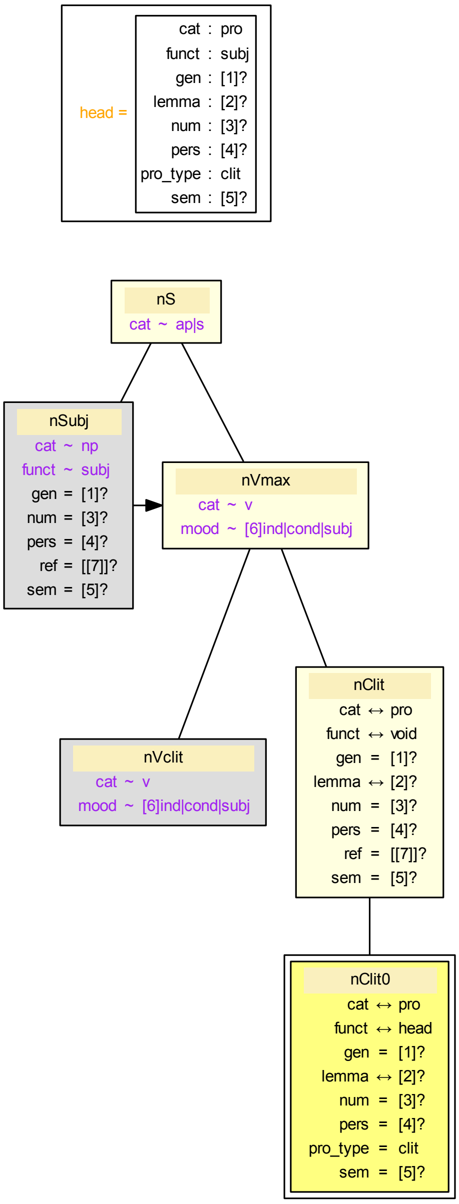

Figure 6.5: PTD defined by the SubjectClitic class

The behaviour of subject clitics is very particular, which is expressed with a specific class, the SubjectClitic class. This class inherits the FullConstituentClitic class and defines the PTD shown on Figure 6.5. Node *nSubj*, which is a renaming of node *nConst* from the FullConstituentClitic class, represents the subject noun phrase in a canonical position. Features `cat` and `funct` attached at this node are virtual because they can be saturated in different ways.

Two sub-classes inherit the SubjectClitic class: StandardSubjectClitic and CeSubjectClitic. The first one corresponds to clitics *il* and *on*, used as actual subjects, which is illustrated with Sentences (6.5) and (6.6), and the second one corresponds to clitic *ce*, which is illustrated with Sentence (6.8). We must deal with clitic *ce* separately because the only possible verb that accepts it as subject is *être*, even if this verb can be modified by modal auxiliaries.

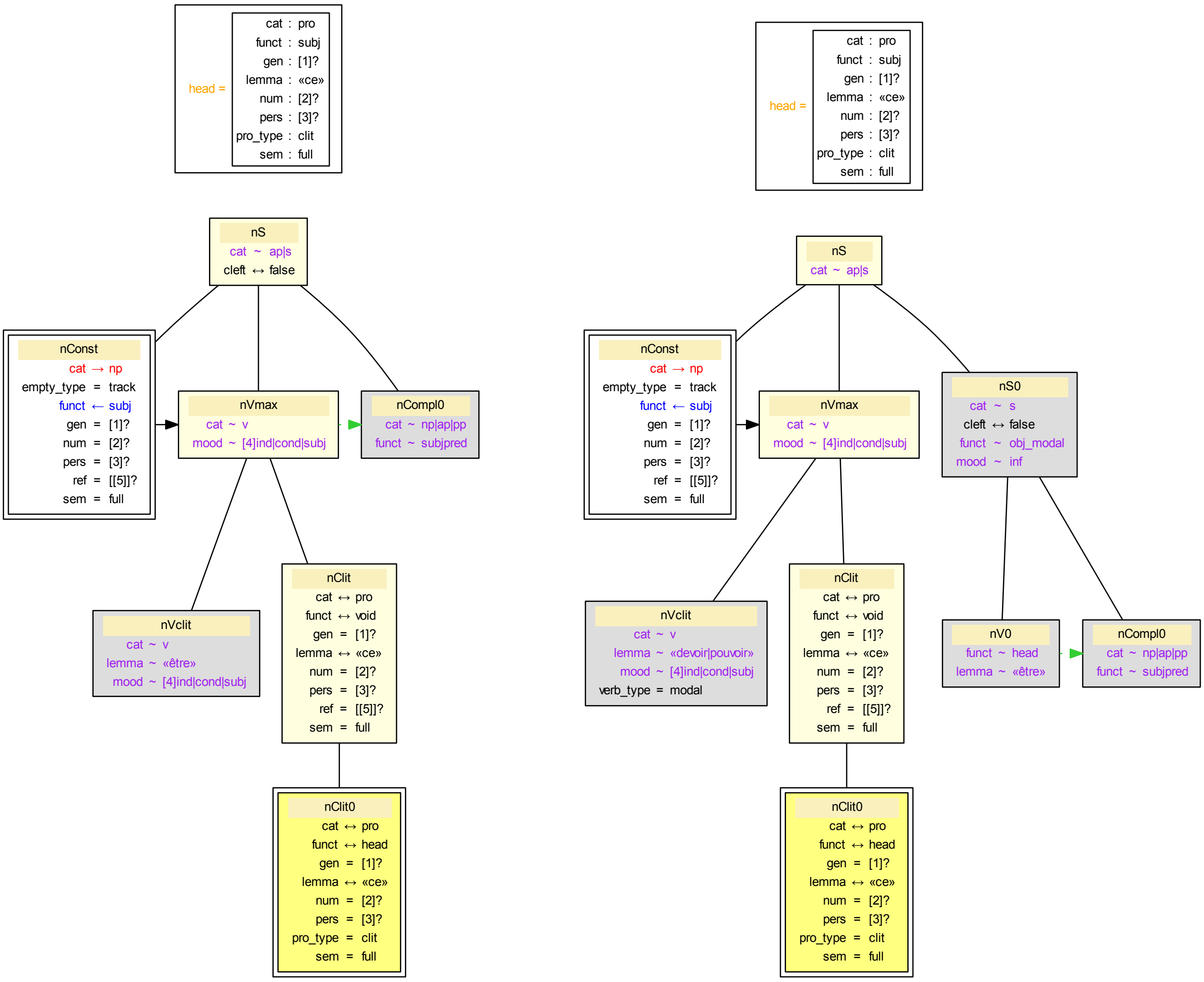

Figure 6.6: Standard PTDs defined by the CeSubjectClitic class

With respect to the SubjectClitic class, StandardSubjectClitic makes node *nSubj* empty, because the clitic pronoun is the only subject of the verb and *nSubj* is a trace of its canonical position. Since *nSubj* is the actual subject, it carries the polarised features `cat → np` and `funct ← subj`. Then, the class assigns the features `sem = full` and `det_type = X` to *nSubj* if the clitic is a personal pronoun (Example (6.6)) and the feature `sem = void` if it is an impersonal pronoun (Example (6.5)). Value `X` depends on the clitic.

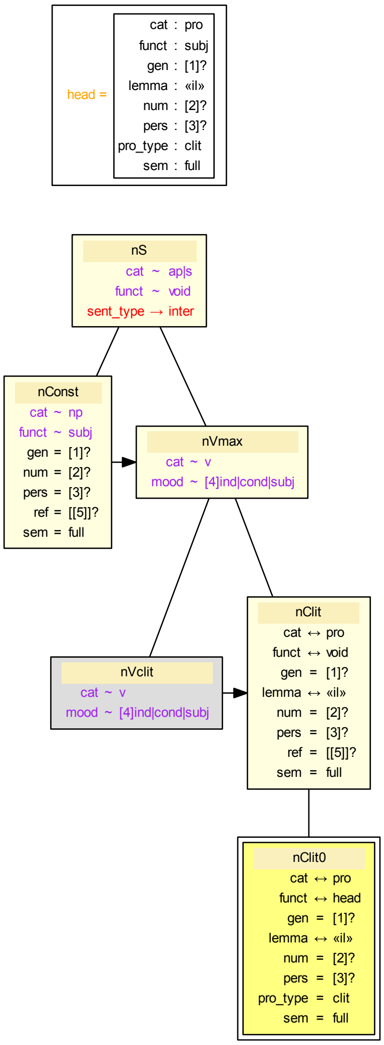

Figure 6.7: EPTD defined by the PROclit-subj-repeat class

The CeSubjectClitic class defines eight PTDs but two of them are used to build dislocated clauses and their description is postponed to section 10.5 of chapter 10. Then, four of the remaining PTDs are used to build cleft clauses and their description is postponed to section 10.4 of chapter 10. The two remaining PTDs are shown in Figure 6.6. The main difference with the StandardSubjectClitic class is that clitic *ce* is necessarily the subject of verb *être*. There are two PTDs because the link between *ce* and *être* can be direct (left PTD) or realised through modal auxiliaries (right EPTD). Example (6.8) illustrates the first case.

The two classes StandardSubjectClitic and CeSubjectClitic are gathered in the disjunction ActualSubjectClitic class. According to the position of the clitic, the ActualSubjectClitic class is divided into two sub-classes: PROclit-subj-decl and PROclit-subj-inter according to the position of the clitic with respect to the verb. Put before the verb as in Sentence (6.5), it expresses that the corresponding sentence is declarative. Put after the verb as in Sentence (6.6), it expresses that the sentence is interrogative.

In both cases, the clitic pronoun plays the role the actual subject.There is another case, where the subject clitic put after the verb is a repetition of the actual subject, which is a noun phrase put before the verb, as Sentence (6.7) illustrates it. The PROclit-subj-repeat implements this case and it generates the EPTD shown in Figure 6.7. The class inherits the SubjectClitic class and adds no polarised feature to node *nSubj*; it only constrains this node to be a full constituent. An effect of subject inversion is that the sentence takes an interrogative form, which is expressed with the positive feature `sent_type` → `inter`.

6.2.3 Verb complement clitic pronouns

Clitic pronoun can be verb complements in various circumstances. The sentences below give different examples of complement clitics.

(6.9) *Jean* **la** *voit .*
Jean her sees .
Jean sees her.

(6.10) *Jean* **y** *a fait venir Marie .*
Jean there made come Marie .
Jean made Marie come there.

(6.11) *Jean* **se** *fait emmener par Marie .*
Jean himself makes take by Marie .
Jean makes himself take by Marie.

(6.12) *Jean fait* **se** *rencontrer les ingénieurs .*
Jean makes meet the engineers .
Jean makes the engineers meet.

(6.13) *Les ingénieurs* **se** *parlent .*
The engineers themselves tell .
The engineers tell themselves.

(6.14) *Jean* **y** *rencontre* *Marie* .
Jean there meets Marie .
Jean meets Marie there.

(6.15) *Jean* **s'** *est acheté* *une* *voiture* .
Jean himself bought a car .
Jean bought a car for himself.

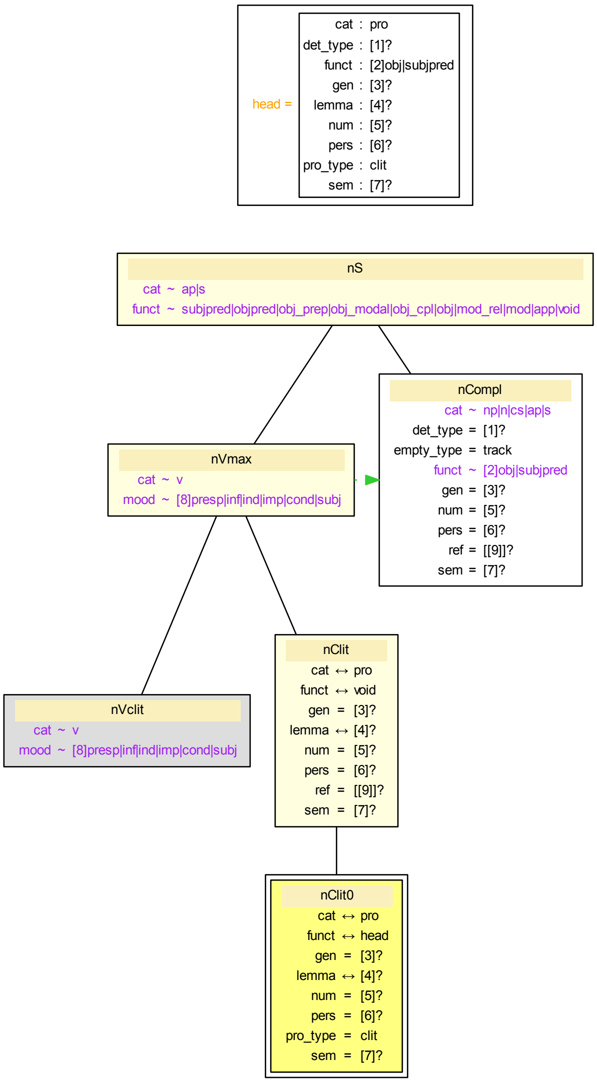

Figure 6.8: PTD defined by the VerbComplementClitic class without clitic climbing in a clause that is not a complement of a causative auxiliary

The PTDs expressing the fact that a clitic pronoun is a verb complement are defined by the VerbComplementClitic class, which has the syntactic function of the complement as a parameter. This class inherits the FullConstituentClitic class, renaming node *nConst* as *nCompl*. It generates 8 PTDs, which correspond to three possible contexts:

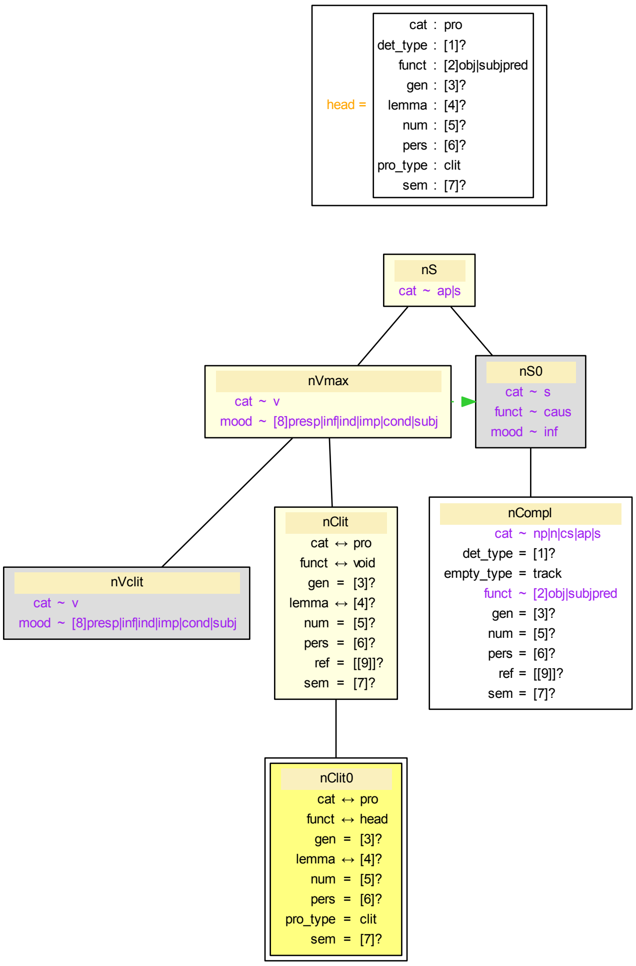

Figure 6.9: PTD defined by the VerbComplementClitic class with clitic climbing in a causative construction

- The clitic pronoun is a complement of the verb that it cliticizes in a clause that is not the complement of a causative auxiliary (see Examples (6.9), (6.13), (6.14) and (6.15)). Figure 6.8 shows one of the three PTDs expressing this case, the PTD for clitics that are direct complements of the cliticized verb. The trace *nCompl* of the clitic is an immediate sub-constituent of *nS*, because it is a complement of the cliticized verb. Feature `funct` of node *nS* expresses the fact that the node represents a clause that is not complement of a causative auxiliary. If the causative auxiliary is a compound verb with a tense auxiliary, the clitic is adjoined to the tense auxiliary, which is represented with node *nVclit*. If the causative auxiliary is a simple verb (Examples (6.9) and (6.13)), the clitic is adjoined to it and is represented with node *nVclit*.

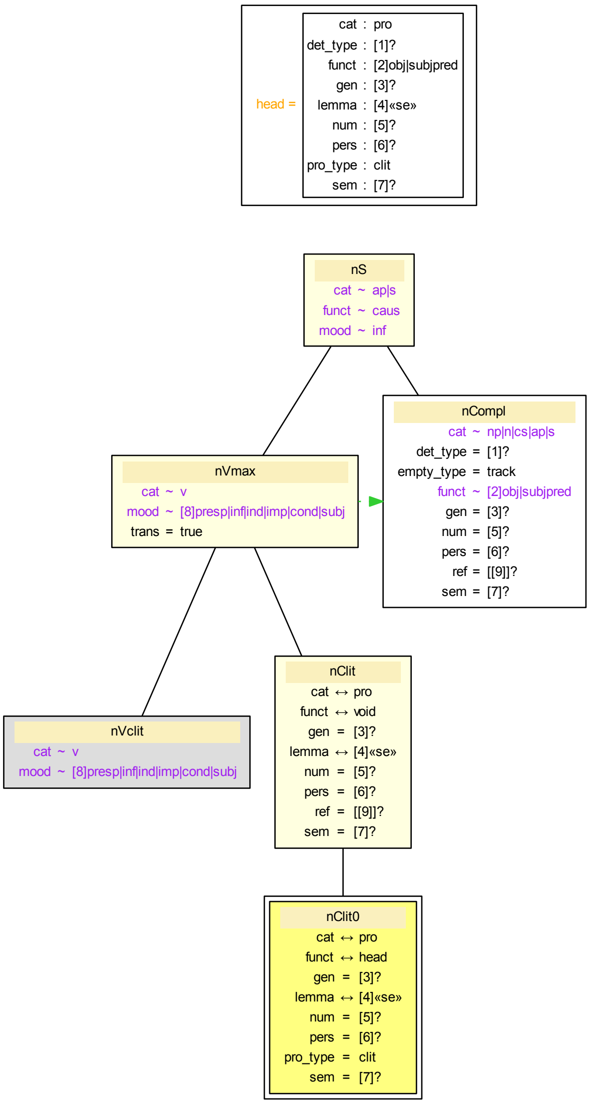

Figure 6.10: PTD defined by the VERBCOMPLEMENTCLITIC class in a causative construction for a reflexive clitic that stays in the clause complement of the causative auxiliary

- The clitic pronoun is a complement of an infinitive that is itself the complement of a causative auxiliary. In this case, the clitic climbs to the causative auxiliary. Examples (6.10) and (6.11) illustrate this case and Figure 6.9 shows one of the three PTDs corresponding to this case, the PTD for clitics that are direct complements of the cliticized verb. Node *nS0* represents the infinitive that is complement of a causative auxiliary. As in the previous case, the position of the clitic depends whether the causative auxiliary is a compound or a simple verb.

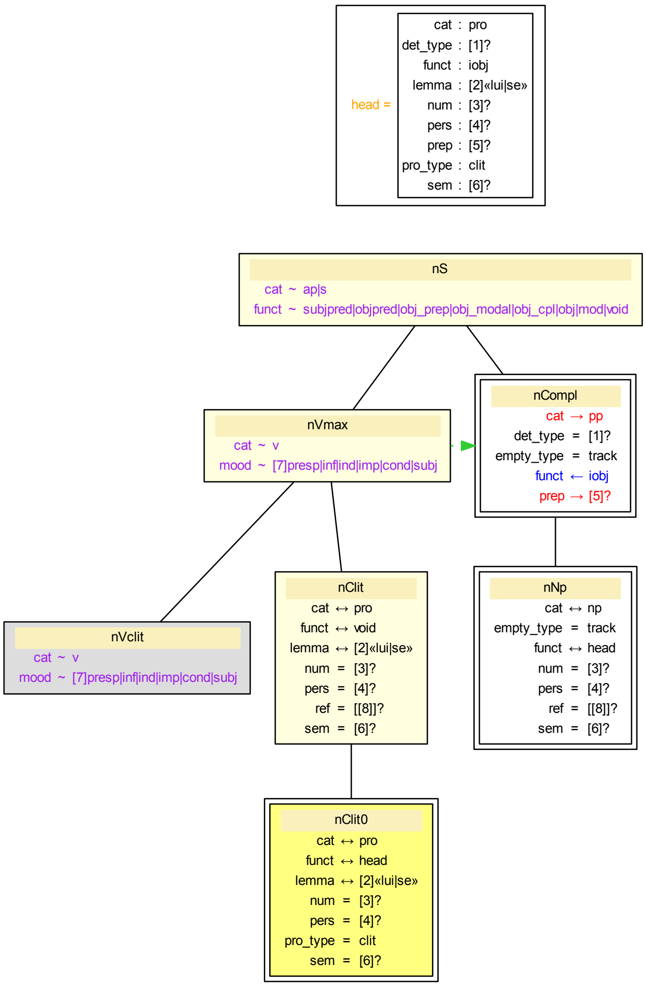

Figure 6.11: PTD defined by the IndirectVerbComplementClitic class for clitic pronouns that are indirect complements required by verbs

- The clitic pronoun is reflexive, it is a complement of an infinitive and it refers to the subject of the infinitive. Moreover, the infinitive is the complement of a causative auxiliary. In this case, the clitic does not climb to the causative auxiliary. Example (6.12) illustrates this case and Figure 6.10 shows one of the two corresponding PTDs.

Among complement clitics, one can distinguish complement clitics which are required by the verb from adjunct clitics. The first case is expressed with two classes: the DirectVerbComplementClitic class illustrated with Sentences (6.9), (6.11) and (6.12), and the IndirectVerbComplementClitic class illustrated with Sentences (6.10) and (6.13).

The two classes inherit the VerbComplementClitic class while adding polarised features to express that the clitic is an obligatory complement. The DirectVerbComplementClitic class adds features `cat` → `np|n|cs|ap|s` and `funct` ← `obj|subjpred` to node *nCompl*. The IndirectVerbComplementClitic class adds features `cat` → `pp`, `funct` ← `iobj` and `prep` ← `X`, where `X` depends on the clitic, to node *nCompl*. Figure 6.11 shows one of the 5 PTDs defined by the IndirectVerbComplementClitic. The 5 cases correspond to the different values of two parameters: the situation of the clitic with respect to a possible causative construction and the variability of its morphology.

The DirectVerbComplementClitic and IndirectVerbComplementClitic classes are then gathered in their disjunction RequiredVerbComplementClitic.

Now, for clitics that are verb adjuncts, there is a similar difference between reflexive (*se*) and non reflexive clitics (*lui, y, en*). This difference is expressed with two classes: NonReflexiveAdjunctVerbComplementClitic and ReflexiveAdjunctVerbComplementClitic. Both inherit the VerbComplementClitic class to which they add saturated features.

Then, on the one hand, the RequiredVerbComplementClitic and NonReflexiveAdjunctVerbComplementClitic classes are gathered in their disjunction NonReflexiveVerbComplementClitic to express all situations of non reflexive clitics that are verb complements.

On the other hand, the RequiredVerbComplementClitic and ReflexiveAdjunctVerbComplementClitic classes are gathered in their disjunction ReflexiveVerbComplementClitic to express all situations of reflexive clitics that are verb complements. With respect to the non reflexive case, this class adds a saturated feature `refl` ↔ `arg` to the verb after cliticization, and it makes node *nClit*, representing the clitic, co-referent of the subject of the cliticized verb.

Figure 6.12 shows the PTD defined by the ReflexiveVerbComplementClitic that is used to parse Sentence (6.15). The figure shows a constraint on the syntactic function of the main sentence, expressed in the feature `funct` of node *nS*. This feature cannot take the value `caus`, which means that the sentence with the cliticized verb cannot be the object of a causative auxiliary, because in this case, the clitic must rise to the causative auxiliary. node *nVmax* carries a saturated feature `refl` ↔ `arg` to express that the verb has a reflexive clitic which is an argument. This argument is a dative modifier of the verb, which is expressed in the trace, which is the subtree rooted at node *nConst*.

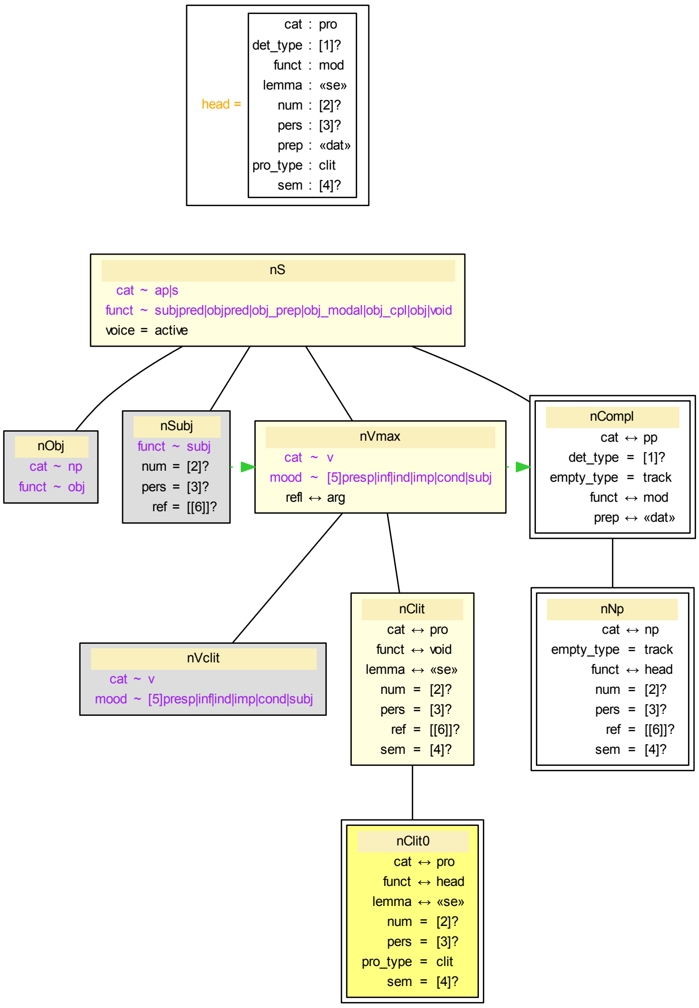

Figure 6.12: PTD defined by the ReflexiveVerbComplementClitic class and used for the parsing of Sentence (6.15)

6.2.4 Noun complement clitic pronouns

The *en* clitic pronoun can play the role of a noun complement as the following examples show it.

(6.16)	*Jean*	**en**	*connaît*	*la*	*couleur*	*.*
	Jean	of it	knows	the	color	.

Jean knows the color of it.

(6.17) *Jean* **en** *connaît* *le* *résumé* *de* *la* *décision* *.*
Jean of him knows the abstract of the decision .
Jean knows the abstract of the decision of him.

(6.18) *Jean* **en** *fait* *publier* *les* *conclusions* *.*
Jean of it makes publish the conclusions .
Jean has the conclusions published.

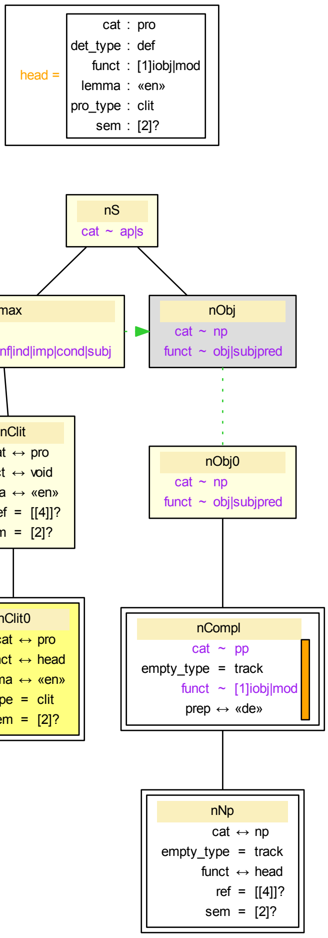

Figure 6.13: PTD defined by the NounComplementClitic class

The basic class is NounComplementClitic. It generates two PTDs according to the presence or not of a causative auxiliary. This distinction comes from the fact that a causative construction entails clitic rising as in Sentence (6.18). In this example, clitic *en* rises from the infinitive *publier* to the auxiliary *fait*. Figure 6.13 shows the case of the absence of causative construction. The trace of the noun complement is represented with an empty subtree rooted at node *nCompl*. Its mother constituent is a noun phrase represented with node *nObj0*. This noun phrase is more or less deeply embedded in another noun phrase which has a function of object or predicate and is represented with node *nObj*. Hence, an underspecified dominance relation between node *nObj* and node *nObj0*.

In Sentences (6.16) and (6.18), nodes *nObj* and *nObj0* are merged and they respectively represent the phrases *la couleur* and *les conclusions*. In Sentence (6.17), node *nObj*, which represents *le résumé de la décision*, strictly dominates node *nObj0*, which represents *la décision*.

The NounComplementClitic class is specialised in two classes, the ObligatoryNounComplementClitic class, when the clitic is a required complement as in Sentences (6.17) and (6.18), and the OptionalNounComplementClitic class, when the clitic is an adjunct as in Sentence (6.16). They only differ in the polarities attached at node *nCompl*: in the former, there are the polarised features `cat` → `pp`, `prep` → `de` and `funct` ← `iobj`; in the latter, there are the saturated features `cat` ↔ `pp`, `prep` ↔ `de` and `funct` ↔ `mod`.

6.2.5 Position of clitic pronouns according to the type of the context clause

According to the type of the context clause (declarative, interrogative, imperative,negative), the clitic pronouns are put before or after the verb. Here are examples illustrating this phenomenon.

(6.19) *Jean* **y** *pense* *.*
Jean it thinks .
Jean thinks it.

(6.20) *Penses*-**y** *!*
Think it !
Think it!

(6.21) *N'***y** *pense* *pas* *!*
it think not !
Do not think it!

(6.22) *Donne* **les** **lui** *!*
Give them him !
Give him them!

The starting common class is the ComplClitic class, which is a disjunction of the NonReflexiveVerbComplementClitic ReflexiveVerbComplementClitic, ObligatoryNounComplClitic, OptionalNounComplClitic, AffixClitic and EmptyReflexiveClitic classes.

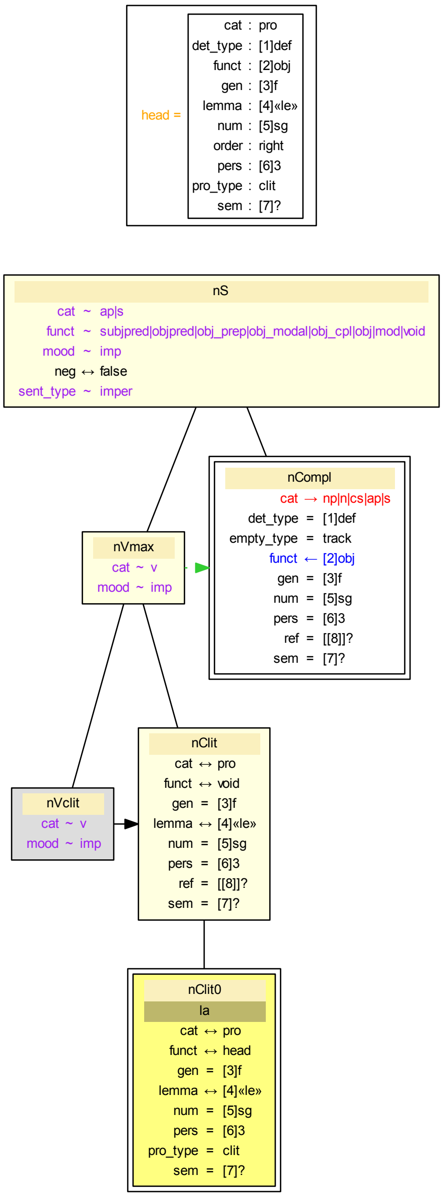

Figure 6.14: EPTD defined by the PROclit-compl-imper-pos class

Then, the ComplCitic class is specialised in three sub-classes:

- the PROclit-compl-decl-inter class for interrogative and declarative clauses, with the clitic pronoun before the verb, as in Example (6.19),
- the PROclit-compl-imper-pos class for imperative positive clauses, with the clitic

pronoun after the verb, as in Example (6.20),

- the PROclit-compl-imper-neg class for imperative negative clauses, with the clitic pronoun before the verb, as in Example (6.21).

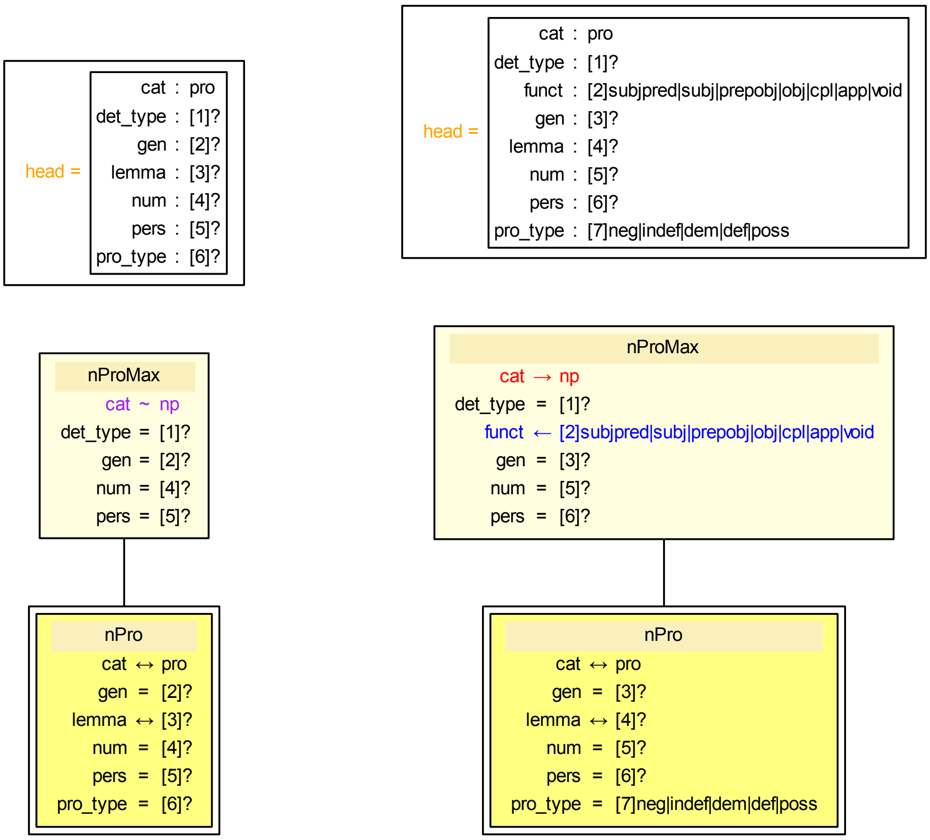

Figure 6.15: PTDs defined by the Pronoun and ArgumentPronoun classes

A difficulty arises in the second case when two clitics are present, as in Sentence (6.22). Hence, the PROclit-compl-imper-pos class is more complicated. It distinguishes the PTD attached at the first clitic from the EPTD attached at the second clitic. Each clitic must verify the absence of a negation. The first one, represented with the EPTD of Figure 6.14, add a saturated feature `neg` ↔ `false` to node *nS* representing the sentence headed by the cliticized verb. The second one brings a virtual feature `neg` ∼ `false`, which will be saturated by the feature brought by the EPTD of the first clitic.

When a verb is equipped with several clitics, their order is ignored by our grammar. Thus, it parses the wrong sentence **je lui le donne* in the same way as the acceptable sentence *je le lui donne*.

6.3 Disjunctive pronouns

The common skeleton for the EPTDs of disjunctive pronouns (by opposition to clitic pronouns) is defined by the Pronoun class and shown on the left of Figure 6.15. The ArgumentPronoun class is a refinement of the Pronoun class for disjunctive pronouns playing the role of an argument. Figure 6.15 shows the corresponding PTD on the right. The only difference with respect to the Pronoun PTD lies in the addition of polarised features `cat` → `np` and `funct` ← `app`|`cpl`|`obj`|`prepobj`|`subj`|`subjpred`|`void`.

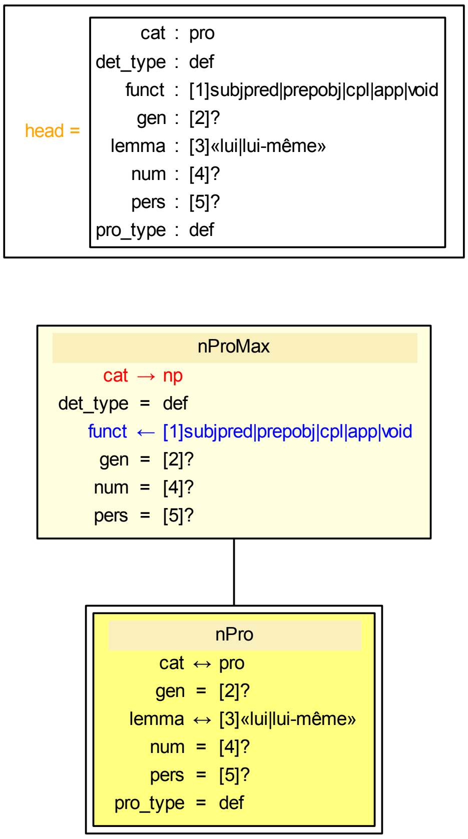

Figure 6.16: EPTD defined by the PROPers class for a standard use

Then, this class is specialized in seven terminal sub-classes:

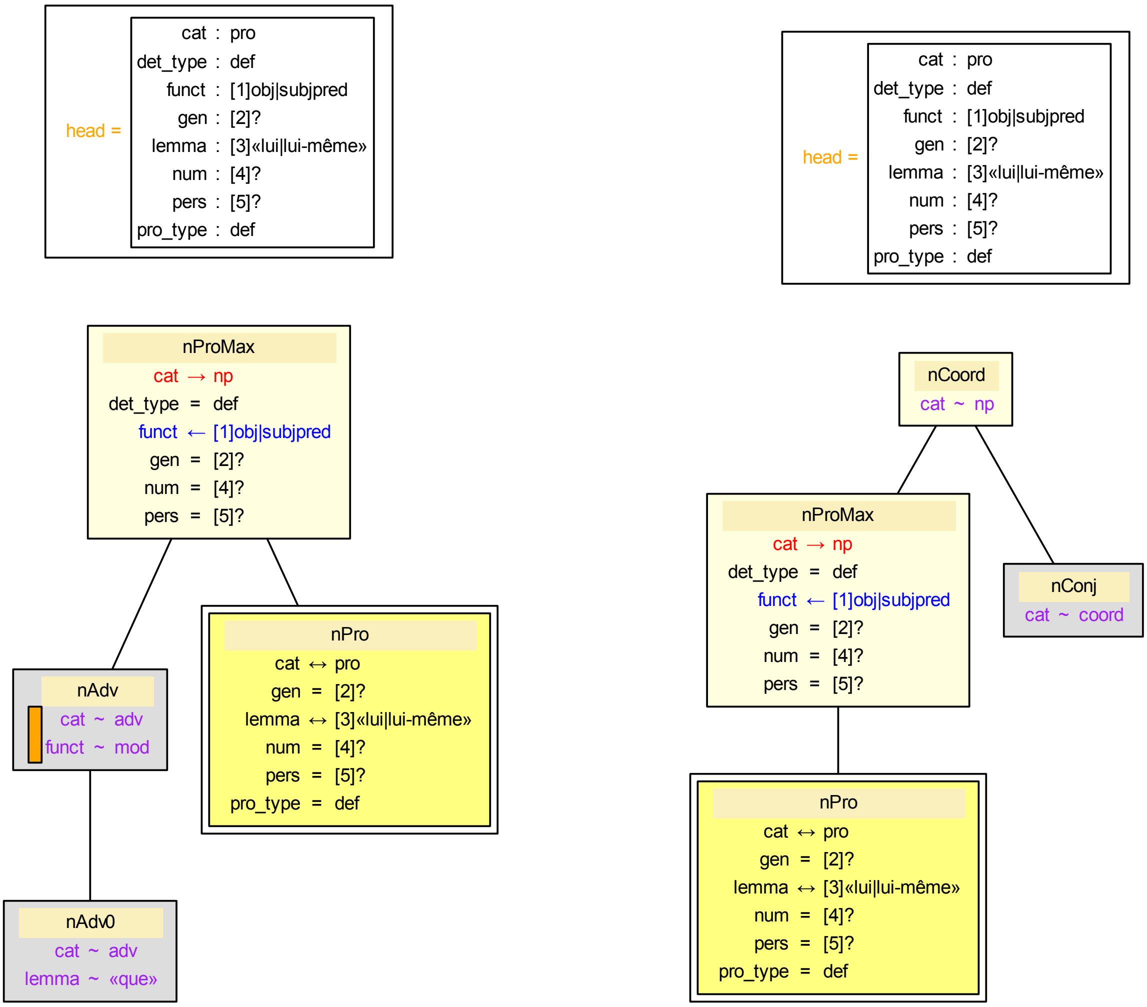

Figure 6.17: EPTDs defined by the PROPERS class for a use with a negation and a coordination

- the PROPERS class dedicated to disjunctive personal pronouns that are not reflexive,
- the PROREFL class dedicated to disjunctive personal pronouns that are reflexive,
- the PRODEM class dedicated to demonstrative pronouns,
- the PROPOSS class dedicated to possessive pronouns,

- the PROindef class dedicated to indefinite pronouns,
- the PROmod_S1 class dedicated to indefinite pronouns used as sentence modifiers,
- the PROneg class dedicated to negative pronouns.

The following examples illustrate the use of non reflexive personal pronouns.

(6.23) *Jean* *travaille* *pour* **lui** *.*
Jean works for himself .
Jean works for himself.

(6.24) *Jean* *ne* *connaît* *que* **lui** *.*
Jean knows only him .
Jean knows only him.

(6.25) **Lui** *et* **moi** *venons* *demain* *.*
He and me are coming tomorrow .
He and me are coming tomorrow.

The PROpers class takes the three cases illustrated with the three sentences above into account. Inheriting the ArgumentPronoun class, it defines three EPTDs.

The first EPTD, presented on Figure 6.16 and illustrated with Sentence (6.23), corresponds to the use of personal pronouns as noun phrases but with restricted functions. For instance, they cannot be used as direct objects of verbs.

The second EPTD, presented on the left of Figure 6.17 and illustrated with Sentence (6.24), corresponds to the use of personal pronouns as direct objects or predicate complements but modified with the negation adverb *que*.

The third EPTD, presented on the right of Figure 6.17 and illustrated with Sentence (6.25), corresponds to the use of personal pronouns as conjuncts in coordination of noun phrases.

Reflexive personal pronouns have similar constraints illustrated with the following examples.

(6.26) *On* *travaille* *pour* **soi** *.*
One works for himself .
One works for himself.

(6.27) *On* *est* *toujours* **soi-même** *.*
One is always himself .
One is always himself.

(6.28) *On* *travaille* *mieux* *pour* *un autre* *que* **soi** *.*
One works better for someone else than himself .
One works better for someone else than himself.

(6.29) *On* *peut* *le* *faire* **soi-même** *.*
One can it do himself .
One can do it himself.

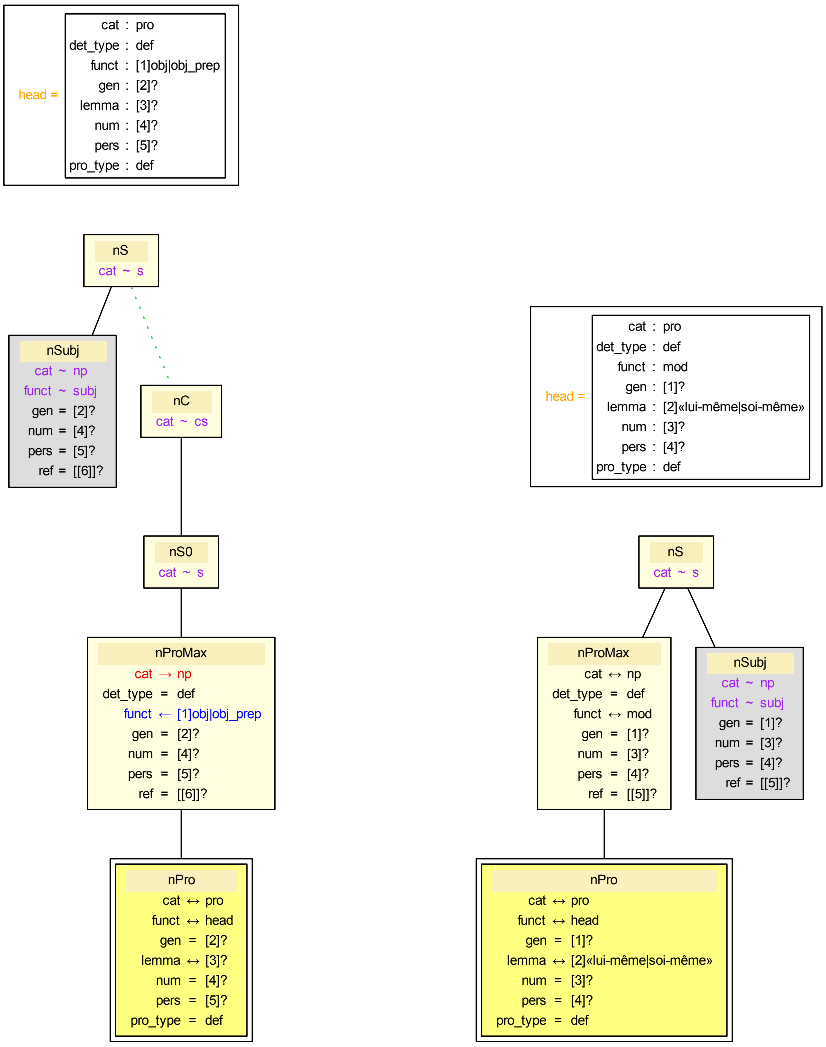

Figure 6.18: Two of the five EPTDs defined by the PROREFL class

The PROREFL class is not exactly a specialisation of the ARGUMENTPRONOUN class because the reflexive pronoun *soi-même* is not always used as an argument but sometimes it plays the role of a modifier as in Sentence (6.29). Therefore, the PROREFL class directly inherits the PRONOUN class and it generates five EPTDs. Two of them are shown on

Figure 6.18. For both of them, node *nSubj* represents the subject of the clause which is the antecedent of the reflexive pronoun.

From the left to the right, they respectively correspond to Sentences (6.28) and (6.29). A first difference lies in the features of node *nProMax*: in the left PTD, *nProMax* is an argument, which is expressed with the polarised features `cat` and `funct`; in the right one, node *nProMax* has all its features saturated because the pronoun is a modifier of the sentence and not a required argument.

A second difference lies in the position of the pronoun in the syntactic tree with respect to the subject that it refers. In the right EPTD, both are sub-constituents of the same sentence, whereas in the left EPTD, the pronoun is embedded in a comparative clause represented by node *nC*, which is itself embedded in the clause for which node *nSubj* represents the subject. In Example (6.28), node *nC* represents the clause with ellipsis *que soi*.

The PROdem, PROposs and PROindef classes are just copies of the ArgumentPronoun class with addition of specific feature to the interface.

The PROneg class concerns negative pronouns, as the following sentences illustrate it.

(6.30) *Jean ne peut arriver à voir* **personne** *.*
Jean can succeed to see nobody .
Jean cannot succeed in seeing somebody.

(6.31) *Jean ne travaille avec l'appui de* **personne** *.*
Jean works with the support of nobody .
Jean works with the support of nobody.

(6.32) **Nul** *ne le connaît .*
Nobody him knows .
Nobody knows him.

These examples show that the position of a negative pronoun with respect to the correlated particle *ne* is flexible, which is expressed in the PROneg class related to negative pronouns. This class generates two EPTDs, shown on Figure 6.19. Nodes *nS1* on the left EPTD and *nS* on the right EPTD represent the clause which is the scope of the negation: it has the verb carrying the particle *ne* as its head. There are two EPTDs because of the following alternative:

- The negative pronoun is in the same clause, which is expressed by the left EPTD and illustrated by Sentences (6.31) and (6.32). It is more or deeply embedded in a noun or prepositional phrase, which is an immediate sub-constituent of the clause and represented with node *nArg*. Hence, there is an underspecified dominance relation from node *nArg* to node *nProMax*, which represents the maximal projection of the pronoun. For Sentence (6.31), node *nArg* represents the phrase *avec l'appui de personne* and it strictly dominates node *nProMax*, which represents *personne*. For Sentence (6.32), node *nArg* and node *nProMax* are merged to represent *nul*.

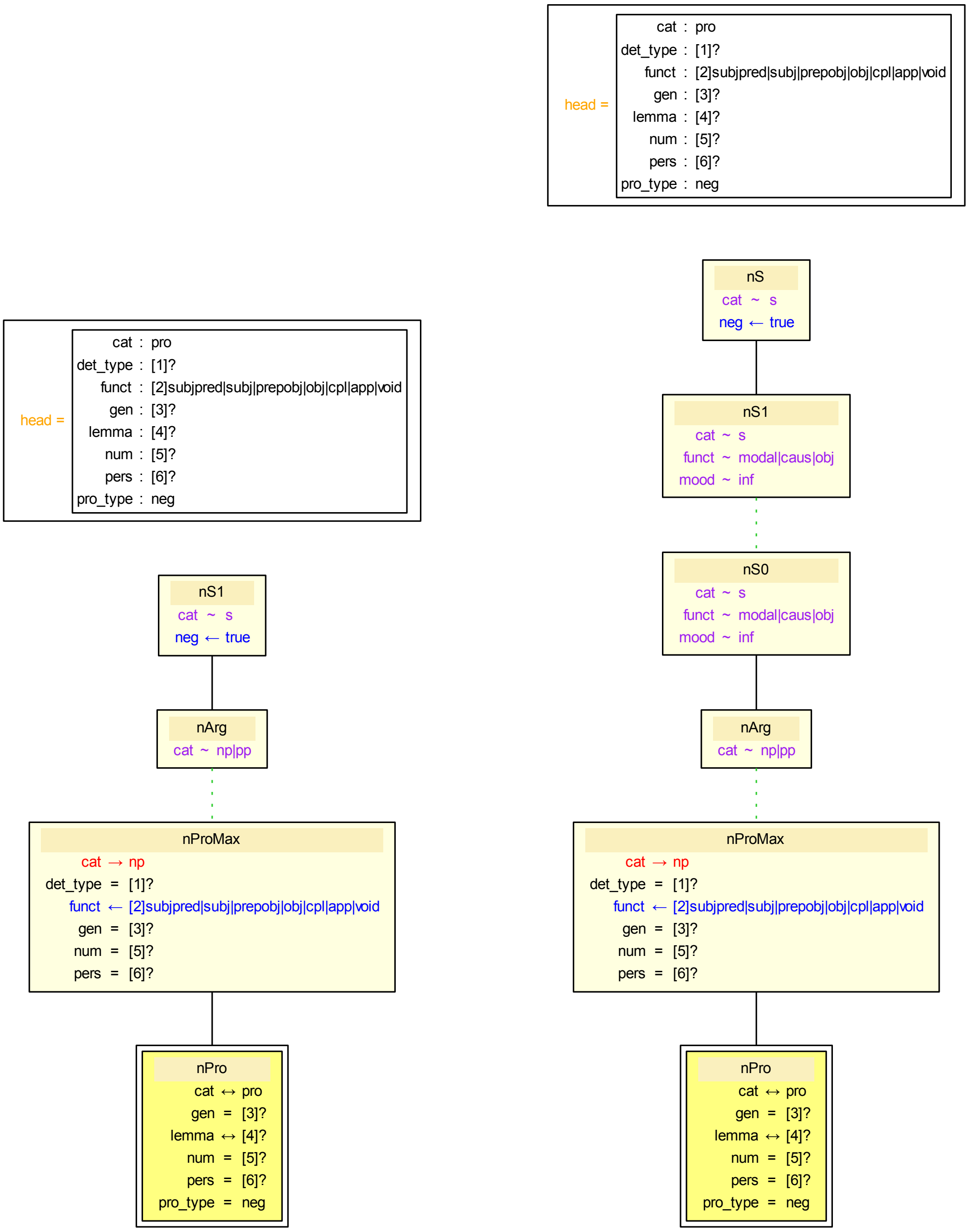

Figure 6.19: EPTDs defined by the PRONEG class

- The negative pronoun is in an infinitive, which is more or less deeply embedded in the scope clause. It is expressed by the right EPTD and illustrated by Sentence (6.30). Node *nS1* represents the infinitive that is an immediate subconstituent of the scope clause. For Sentence (6.30), it corresponds to *arriver à voir personne*. Thus, there is an underspecified dominance relation form node *nS1* to node *nS0*, which represents the phrase *voir personne* in our example. Then, we have another dominance relation from node *nArg* to node *nProMax*, as in the other EPTD, but in Example (6.30), the two nodes merge to represent *personne*.

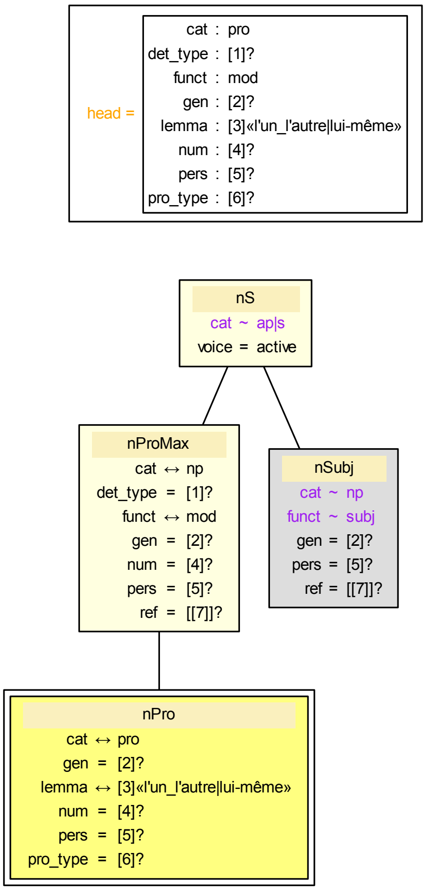

Figure 6.20: EPTD defined by the PROMOD class

Another kind of indefinite pronouns have a specific behaviour. They act as modifiers of sentences co-referring with the subject of theses sentences. Here are examples with such pronouns.

(6.33) *Il* *fait* *tout* **lui-même** *.*
He makes all himself .
He makes all himself.

(6.34) *Les* *étudiants* *s'* *apprécient* **les uns les autres** *.*
The students themselves appreciate each other .
The students appreciate themselves each other.

The class defining the EPTD for indefinite pronouns acting as sentence modifiers, is PROmod_S1. This EPTD is shown on Figure 6.20. Feature `ref` expresses that nodes *nProMax*, representing the maximal projection of the pronoun, and *nSubj*, representing the subject of the sentence or the adjectival phrase, co-refer to the same entity.

Some quantifier pronouns, like *tous* or *chacun*, have the same behaviour but in a more flexible way, because they may co-refer with complements of the verb. They will studied in the next section.

6.4 Quantifier pronouns

Quantifier pronouns, like *tous*, *tout*, *chacun*, *rien*, have a specific syntax, which requires specific classes. First, *tous*, *tout* and *rien* can behave as clitic pronouns, as the following examples show it.

(6.35) *Jean* *n'a* **rien** *compris* *.*
Jean has nothing understood .
Jean has understood nothing.

(6.36) *Jean* *ne* *comprend* **rien** *.*
Jean understands nothing .
Jean understands nothing.

(6.37) *Jean* *a* **tout** *pu* *faire* *réaliser* *par* *son* *frère* *.*
Jean has all can made achieve by his brother .
Jean could have made all to be achieved by his brother.

(6.38) *Jean* *n'a* *pu* **rien** *faire* *réaliser* *par* *son* *frère* *.*
Jean has could nothing made achieve by his brother .
Jean could have made nothing to be achieved by his brother.

(6.39) *Jean* *a* **toutes** *pu* *les* *faire* *examiner* *par* *le* *médecin* *.*
Jean has all could them make examine by the doctor .
Jean could have made all of them to be examined by the doctor.

If Sentence (6.36) does not justify the treatment of *rien* as a clitic, the position of the pronoun before the past participle in other sentences speaks in favour of this treatment.

The CliticQuantifiedPronoun class models this behaviour and generates four PTDs because of the possible combination of two alternatives:

- if the mood of the verb is `pastp` (past participle) or `inf` (infinitive), the pronoun precedes the verb (all examples except (6.36)); if not, the pronoun is put after the verb (Sentence (6.36));
- in presence of a causative or modal auxiliary, the pronoun can rise to the auxiliary (Sentences (6.37), (6.38) and (6.39)).

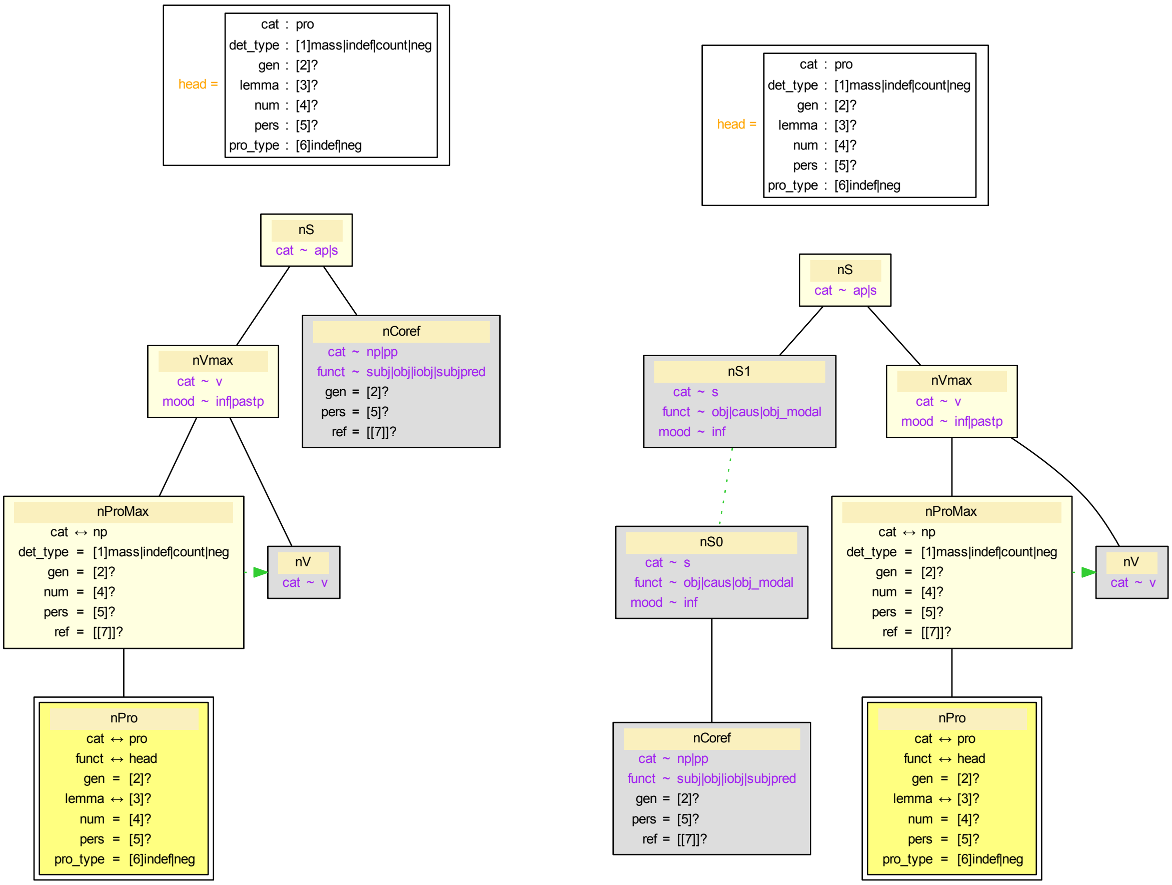

Figure 6.21: Two examples of PTDs defined by the CliticQuantifiedPronoun class

Figure 6.21 shows the cases corresponding to the position of the pronoun before the verb. In both PTDs, the maximal projection of the pronoun, represented with node *nProMax*, adjoins a verb *nV* as a clitic to build node *nVmax* with him.

node *nProMax* co-refers to a node *nCoref* with the help of a feature `ref`. Node *nCoref* may represent an empty trace of the clitic as in the four first examples above. It also can co-refer with an explicit expression as in the last example, where *toutes* co-refers with *les*. Since nodes *nProMax* and *nCoref* co-refer to the same entity, they share their agreement features but they can have different functions given by feature `funct`.

The two PTDs differ in the position of *nCoref*.

- The left PTD represents the configuration without rising of the pronoun, illustrated with Sentence (6.35). Node *nCoref* represents an argument of the verb that is cliticized with the pronoun.

- The right PTD represents the configuration with the rising of the pronoun to a causative or modal auxiliary and it is illustrated with Sentences (6.37), (6.38) and (6.39). Here, node *nCoref* is included in an infinitive represented with node *nS0*. This infinitive can be embedded more or less deeply in the clause, the head verb of which is clitized with the pronoun. This is expressed with an underspecified dominance relation from node *nS1* over node *nS0*. Node *nS1* represents the infinitive which is an immediate sub-constituent of the main clause.For instance, in Sentence (6.39), node *nS1* corresponds to *les faire examiner par le médecin* and node *nS0* to *examiner par le médecin*.

Then, the CliticQuantifiedPronoun class is divided in two more specific classes:

- The DirectComplementQuantifiedPronoun class when the pronoun plays the role of an actual object for the verb, which is illustrated with Sentences (6.35), (6.36), (6.37) and (6.38). In this case, node *nCoref* on Figure 6.21 represents the empty trace of the complement in its canonical position and it carries the polarised features `cat` → `np` and `funct` ← `obj|subpred`.

- The PROtous-mod_S1 class, when the pronoun is *tous* and plays the role of a verb modifier, which is illustrated with Sentence (6.39). In this case, the pronoun co-refers with the subject or the object of the verb. In Sentence (6.39), it co-refers with the object.

The DirectComplementQuantifiedPronoun class itself is specialised in two sub-classes PROtout_V1 and PROrien_V1 . The first one is dedicated to the *tout* pronoun and it is just a copy of the DirectComplementQuantifiedPronoun class.

The second one is dedicated to the *rien* pronoun. It is more complicated because it must express the link with the *ne* particle under the form of the feature `neg` ← `true`, which is attached to the clause that is the scope of the negation.

So three levels can be distinguished, from the most to the less external: the scope of the negation, the scope of the clitized verb and the scope of the verb that has the pronoun as its argument. The three levels can be distinct and all cases still combine with the two possibilities for the position of the pronoun with respect to the verb it cliticizes. Hence, the class generates eight EPTDs.

Figure 6.22 shows two of these EPTDs illustrated with Sentences (6.35) and (6.38). On the right EPTD, nodes *nSne*, *nSne1*, *nS*, *nS1 nS0* represent the following levels illustrated with the following phrases in Example (6.39):

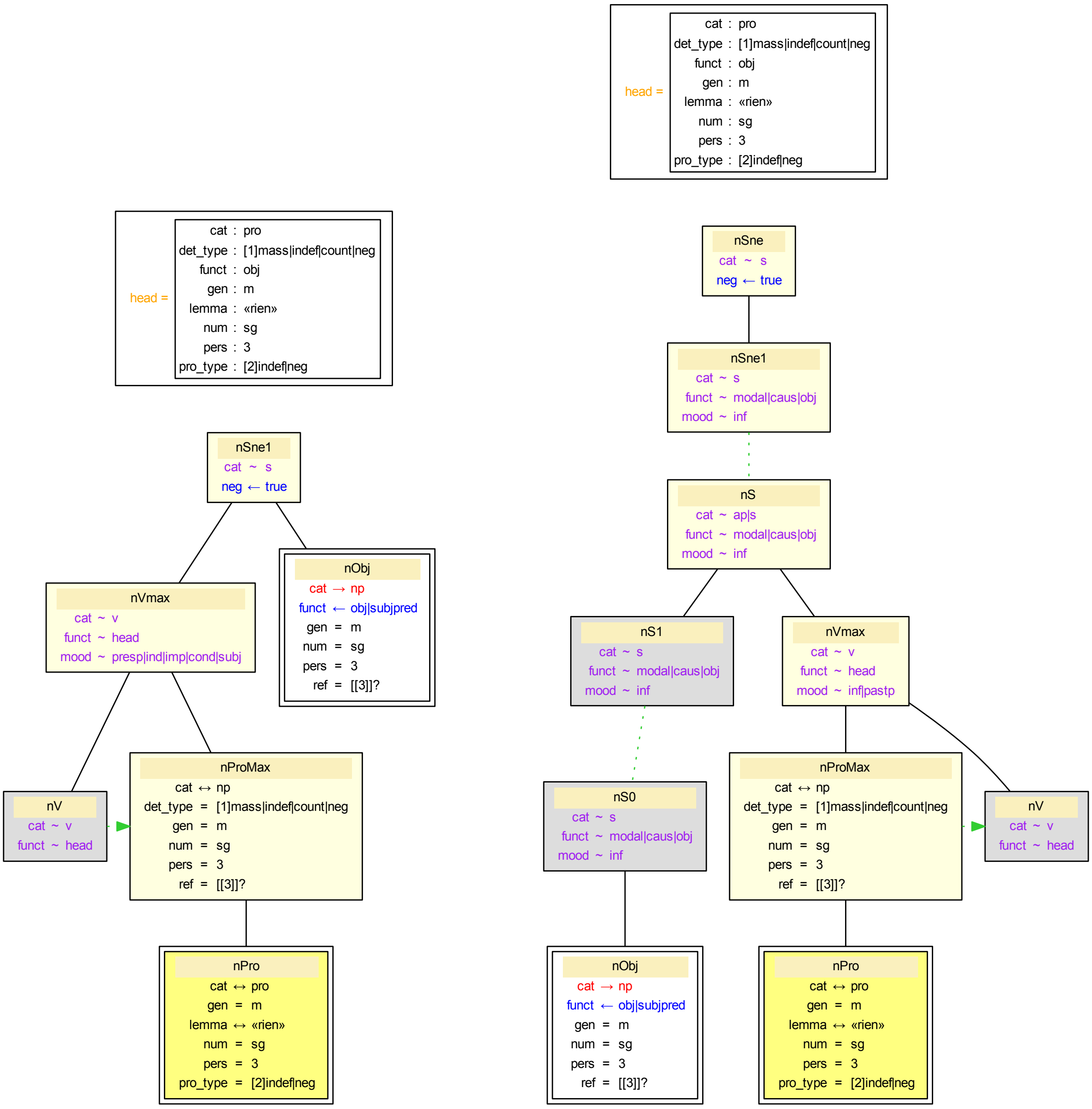

Figure 6.22: Two examples of EPTDs defined by the PRORIEN_V1 class

- *nSne* for the scope of the negation: *Jean n' a pu rien faire réaliser par son frère,*

- *nSne1* for the infinitive that is an immediate sub-constituent: *rien faire réaliser par son frère*,
- *nS* for the scope of the verb cliticized with quantifier pronoun: *rien faire réaliser par son frère*,
- *nS1* for the infinitive that is an immediate sub-constituent: *réaliser par son frère*,
- *nS0* the scope of the verb that has the quantifier pronoun as a complement: *réaliser par son frère.*

Pronouns *tous* and *chacun* behave as sentences modifiers in a similar way as pronouns attached at class PROmod_S1. The first one is used in a more flexible way because it can co-refer with complements but only if these complements are clitic or relative pronouns. So, Sentence (6.40) is grammatical because *tous* co-refers with *leur* but Sentence (6.41) is ungrammatical because *tous* co-refers with *aux enfants.*

(6.40) *Jean leur a* **tous** *donné une pomme .*
Jean them has all given an apple .
Jean has given an apple to all of them.

(6.41) **Jean a* **tous** *donné une pomme aux enfants .*
Jean has all given an apple to the children .
Jean has given an apple to all of the children.

(6.42) *Ils ont mangé une pomme* **chacun** *.*
They have eaten an apple each one .
They have each one eaten an apple.

(6.43) *Les enfants ont* **chacun** *mangé une pomme .*
The children have each one eaten an apple .
The children have each one eaten an apple.

(6.44) *Ils ont mangé* **chacun** *une pomme .*
They have eaten each one an apple .
They have each one eaten an apple.

The PROtous-mod_S1 class models this behaviour, defining eight EPTDs. Figure 6.23 shows the case that the co-referring expression is a complement and it is illustrated with Sentence (6.40). As we can see, node *nCoref* representing the trace of this expression is empty. This constraints entails the failure of parsing for Sentence (6.41).

Pronoun *chacun* is used in a more restricted way because the co-referring expression must be the subject of the cliticized verb, as the three last examples above show it.

Finally, the pronoun *tout* with all its inflected forms behave as a sur-determiner, as the following example illustrates it.

(6.45)	*L'ingénieur*	*accepte*	**toutes**	*les*	*propositions*	*.*
	The engineer	accepts	all	the	proposals	.

The engineer accepts all proposals.

This particular behaviour is modelled with the PROsurdet_NP1 class, which defines the EPTD of Figure 6.24.

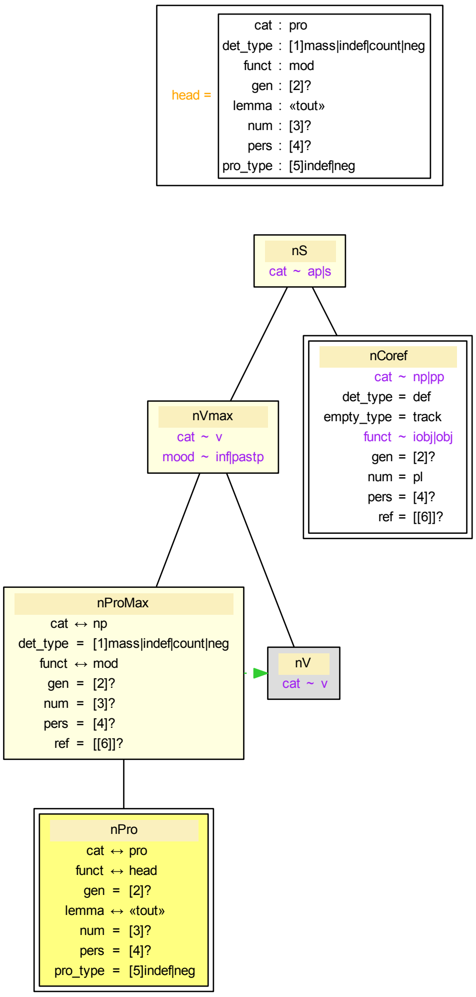

Figure 6.23: EPTDs defined by the PROtous-mod_S1 class

6.5 Pronouns requiring complements

Some demonstrative and indefinite pronouns require prepositional or clausal complements.

6.5.1 Demonstrative and indefinite pronouns with prepositional complements

Demonstrative pronouns, like *celui*, and indefinite pronouns, like *aucun*, *quelqu'un*, *quelques-uns* require a partitive nominal complement introduced with a partitive preposition *de*, *d'entre* or *parmi* as the following sentences illustrate it.

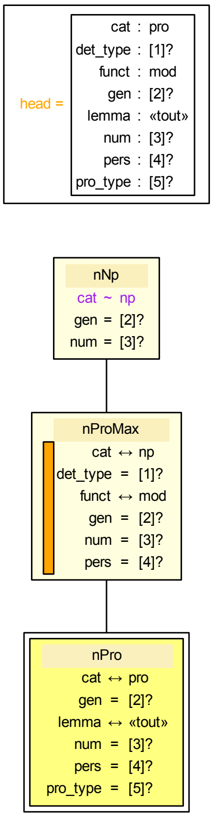

Figure 6.24: EPTDs defined by the PROsurdet_NP1 class

(6.46)	**Celui**	*de*	*Paris*	*vient*	*aujourd'hui*	*.*
	That	from	Paris	is coming	today	.

That from Paris is coming today.

(6.47) *Je* *préfère* **celui** *de* *laine* .
I prefer that with wool .
I prefer that with wool.

(6.48) **Quelques-uns** *d'entre* *mes* *amis* *seront* *présents* .
Someones among my friends will be present .
Someones among my friends will be present.

(6.49) **Aucun** *de* *mes* *amis* *ne sera* *présent* .
Nobody of my friends will be present .
Nobody of my friends will be present.

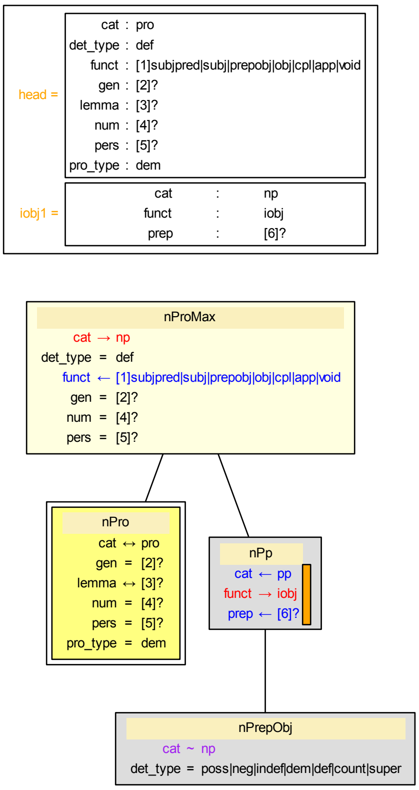

Figure 6.25: EPTD defined by the PRO_PP1NOM class for demonstrative pronouns with a noun phrase complement

(6.50) *Je* *veux* **quelque chose** *de* *facile* .
I want something of easy .
I want something easy.

(6.51)						
Personne	*de*	*sensé*	*ne peut*	*croire*	*cela*	*.*
Nobody	of	sensible	can	believe	that	.

No sensible person can believe that.

When the complement is nominal as in the four first examples, the behaviour of the pronouns is modelled with the PRO_PP1NOM class. When it is adjectival or clausal as in the two examples, it is modelled with PRO_PP1SENT class.

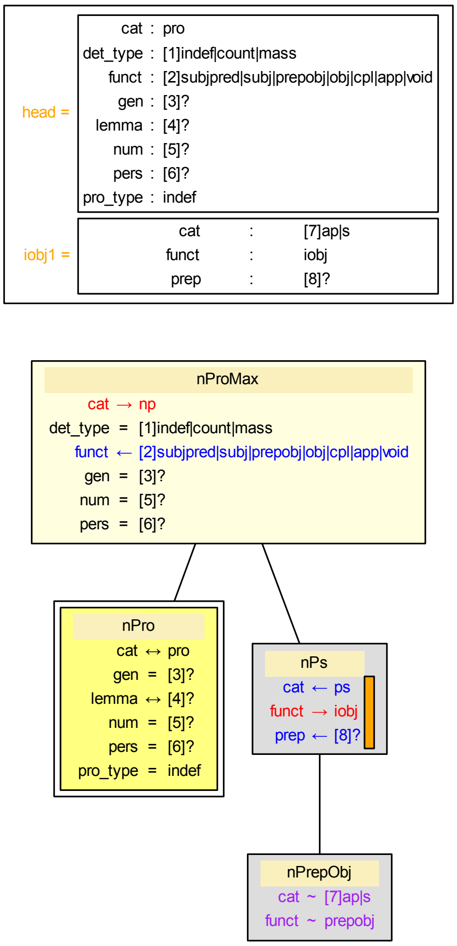

Figure 6.26: EPTD defined by the PRO_PP1SENT class for positive indefinite pronouns

The PRO_PP1NOM class generates eight EPTDs, two for demonstrative pronouns, two for positive indefinite pronouns and four for negative indefinite pronouns. The first EPTD for demonstrative pronouns is shown on Figure 6.25. The only difference with respect to the other EPTD lies in feature `det_type` of node *PrepObj*, the complement introduced with *de*: when the complement is a complete noun phrase as in Sentence (6.46), it has the value `count|def|dem|indef|neg|poss|super`; when the complement is a common noun as in Sentence (6.47), it has the value `voiddet`. The value of the `iobj1.cat` in the interface is consistent with the value of `det_type`.

The PRO_PP1sent concerns indefinite pronouns taking an adjectival complement as in Sentences (6.50) and (6.51). It generates three EPTDs, one for positive indefinite pronouns and two for negative indefinite pronouns. Figure 6.26 shows the EPTD associated with positive indefinite pronouns as Example (6.50) illustrates it.

6.5.2 Demonstrative pronouns with clausal complements

The demonstrative pronouns *ce*, *celui*, *ça* and *cela* have a very specific behaviour illustrated with the following sentences.

(6.52) *Jean connaît* **celui** *qui vient .*
Jean knows that one who is coming .
Jean knows that one who is coming.

(6.53) *Marie croit à* **ce** *que dit Jean .*
Marie believes in that what says Jean .
Marie believes in what Jean says.

(6.54) *Marie sait* **ce** *pour quoi Jean vient .*
Marie knows that for which Jean comes .
Marie knows that for which Jean comes.

(6.55) *Jean s'attend à* **ce** *que Marie vienne aujourd'hui .*
Jean expects that that Marie come today .
Jean expects that Marie comes today.

(6.56) **Cà** *ne m'arrange pas de vous recevoir .*
That does not arrange me to you receive .
That does not arrange me to receive you.

(6.57) *Je trouve* **cela** *dommage que Jean ne vienne pas .*
I find that a pity that Jean does not come .
I find that is a pity that Jean does not come.

As the three first sentences show it, they can require a relative clause, which is modelled with the PROdem_S1rel class, which generates the EPTD shown on the left of Figure 6.27. Node *nS* represents the relative clause that is expected. Node *nProMax* represents the maximal projection of the demonstrative pronoun with the relative clause as a modifier and it behaves as any noun phrase.

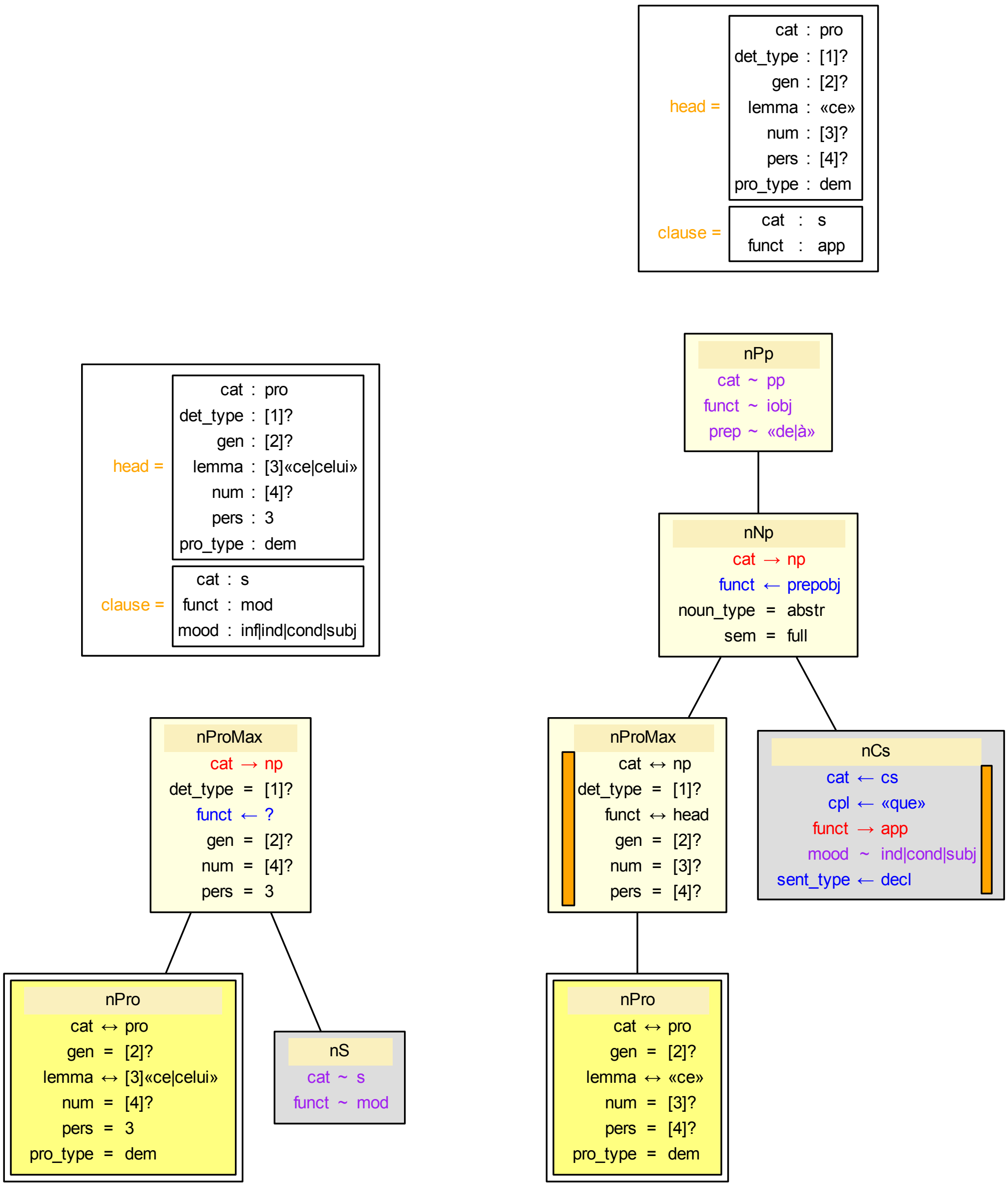

Figure 6.27: The EPTDs defined by the PRODem_S1rel and PROce_CS1fin classes

On the other hand, *ce* can be used with a clause introduced with the complementizer *que*, as Sentence (6.55) illustrates it. This is represented with the PROce_CS1fin class. The class generates the EPTD shown on the right of Figure 6.27. Contrary to the previous case, the maximal projection *nProMax* cannot be used as any noun phrase but it must be the object of prepositional phrase represented with node *nPp*.

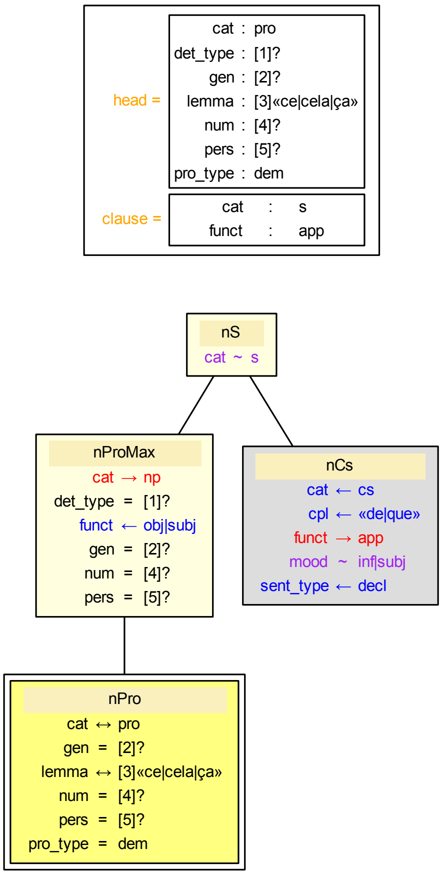

Figure 6.28: The EPTD defined by the PRODEM_CS1APP class

Examples (6.56) and (6.57) illustrate a particular property of some demonstrative pronouns: they can be the subject or the object of a verb to represent a finite or infinitive clause dislocated after the head verb. The PRODEM_CS1APP class models this phenomenon and it defines the EPTD given by Figure 6.28.

Chapter 7

Adjectives

The classes anchored with adjectives are gathered in the Adjective module.

7.1 Interfaces with the lexicon

Adjectives are characterised in the interface with the feature `head.cat = adj`. Their properties are described with the following sub-features of the `head` feature:

- `funct`: it describes the possible syntactic functions among `mod`, `obj`, `objpred`, `obj_prep`, `subjpred`;
- `gen`: it indicates the gender of the adjective with the values `f` and `m`;
- `num`: it gives the number of the adjective, `pl` (plural) or `sg` (singular);
- `order`: when the adjective is attributive, it gives its position with respect to the noun it modifies: `left` or `right`.

Some adjectives require complements, which expressed with features `iobj1`, `iobj2`, according to the number of these complements. Other adjectives include an idea of comparison or consequence in themselves and they require a clause introduced with *que*. In their interface, the properties of this complement are expressed with a feature `arg`

All these complements are described with the following sub-features of `iobj1`, `iobj2`, `arg`:

- `cat`: it indicates the category of the complement: `n` (common noun), `np` (noun phrase) or `s` (sentence);
- `control`: when an adjective takes an infinitive as complement, it indicates the function of argument of the infinitive that is controlled by the subject of the adjective: `obj` (for instance *facile*) or `subj` (for instance *lent*);
- `mood`: when the complement is a clause, it gives the mood of the clause, `inf` (infinitive) or `subj` (subjunctive).
- `prep`: it gives the preposition introducing the complement.

7.2 The attributive and predicate functions of adjectives

Adjectives mainly occur in two syntactic constructions: attributive (Example (7.1)) and predicate (Example (7.2)).

(7.1) *Marie est une femme heureuse .*
Marie is a woman happy .
Marie is a happy woman.

(7.2) *Marie est heureuse de vivre .*
Marie is happy to live .
Marie is happy to live.

7.2.1 Predicate adjectives as complement versus head of clauses

In the predicate construction, an adjective is composed with a verb, which can be interpreted in two manners: the adjective is the head of a clause and the verb (most times the copula) is considered as an auxiliary; or the adjective is a complement of the verb with an attributive function (with respect to the subject or the object of the verb). The first interpretation is justified by some redistributions in the sentence governed by the adjective like in the following examples.

(7.3) *Que Marie dorme est heureux .*
that Marie sleeps is happy .
it is happy that Marie sleeps.

(7.4) *Il est heureux que Marie dorme .*
it is happy that Marie sleeps .
it is happy that Marie sleeps.

At the opposite, possible extractions, as the following examples illustrate it, lead us to consider adjectives composed with a verb as a complement of this verb.

(7.5) *Marie apparaît heureuse de vivre .*
Marie looks happy to live .
Marie looks happy to live.

(7.6) *Comment Marie apparaît-elle ?*
how Mary does-she-look ?
how does Mary look?

(7.7) *Heureuse de vivre, Marie l'est .*
Happy to live, Marie is .
Mary is happy to live.

We have chosen to consider predicate adjectives as complements of the verb to which they refer and in this case, they build an adjectival phrase with their own complements. So in Example (7.5), *heureuse de vivre* is taken as an adjectival phrase complement of the verb *apparaît*.

7.2.2 Left attributive adjectives versus right attributive adjectives

In the attributive construction, adjectives modify common nouns but they do not have the same syntactic properties when they are before the noun they modify (left adjectives) as after the noun (right adjectives).

(7.8) *Marie est une femme heureuse de vivre .*
Marie is a woman happy to live .
Marie is a woman happy to live.

(7.9) *Marie attend un heureux évènement .*
Marie is waiting for an happy event .
Marie is waiting for an happy event.

(7.10) **Marie est une heureuse de vivre femme .*
Marie is a happy to live woman .
Marie is a woman happy to live.

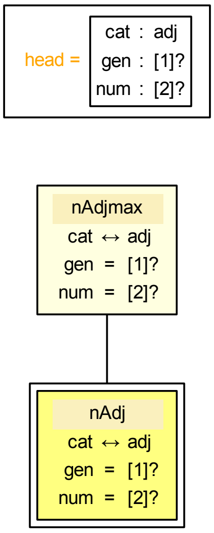

Figure 7.1: PTD defined by the Adjective class

The two first examples show that the same adjective has not the same meaning when it is in left position and when it is in right position but this remark is relevant to semantics, which goes beyond our purpose. From a syntactic point of view, Example (7.10) illustrates the fact that attributive adjectives with complements are always right adjectives. Hence, we represent the syntax of left adjectives differently from right adjectives.

As we will see later, left adjectives combine with the common nouns they modify to build a constituent with the type common noun. Right adjectives build adjectival phrases with their complements and these adjectival phrases combine with the common noun they modify and possibly a determiner to build a noun phrase.

7.2.3 Modelling left attributive adjectives

After this preliminary linguistic discussion, let us enter the modeling of adjectives with IG.

A basic class Adjective expresses the common syntactic properties of all adjectives. It defines the PTD given by Figure 7.1. In this PTD, the anchor node *nAdj* represents the bare adjective. Its mother node *nAdjmax* represents the adjectival kernel constituted of the adjective with its possible modifiers.

The LeftAttributive class is dedicated to left attributive constructions. It inherits the Adjective class and it defines one PTD shown on the left of Figure 7.2. Node *nN* represents the noun modified by the adjective and the mother node *nNmax* represents the noun with its left modifiers.

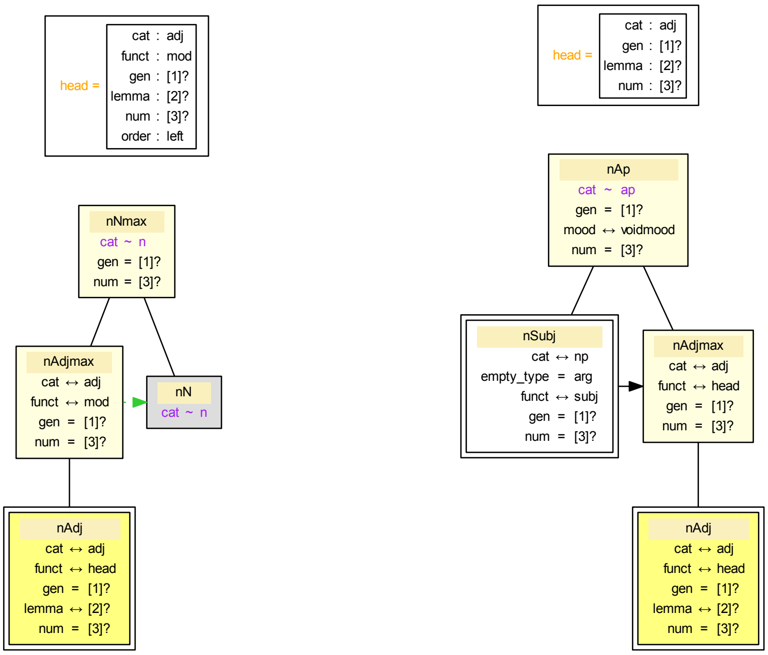

Figure 7.2: PTD defined by the LeftAttributive and AdjectivalPhrase0 classes

7.2.4 Modelling right attributive and predicate adjectives

A common class AdjectivalPhrase0 represents the construction of adjectives as heads of adjectival phrases; which concerns both right attributive and predicate adjectives. This class inherits the Adjective class and it defines the PTD shown on the right of Figure 7.2.

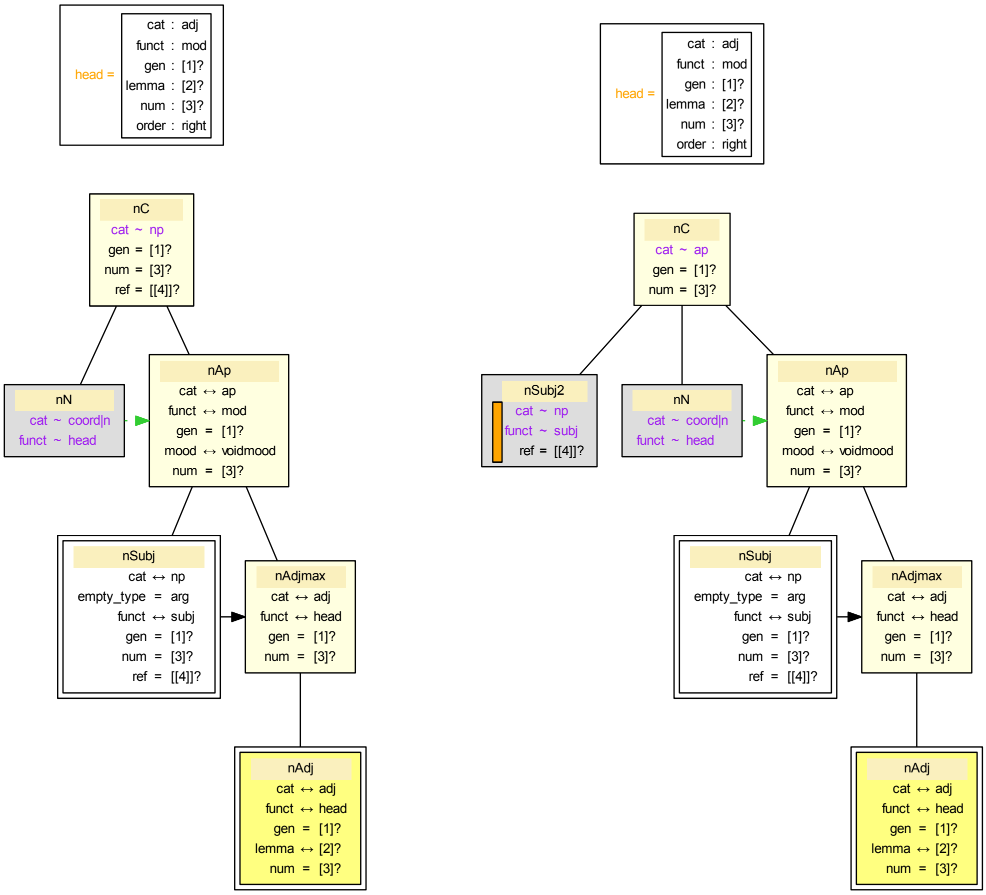

Figure 7.3: PTDs defined by the RightAttributive class

In this PTD, node *nAp* represents the adjectival phrase, which has the adjective as its head, which is expressed with the saturated feature `funct ↔ head`. Like a sentence, an adjectival phrase has a subject represented with node *nSubj*. This node is empty and it agrees in number and gender with the adjective, which is expressed with value sharing for `gen` and `num` features.

The AdjectivalPhrase0 is specialised in two classes: RightAttributive for right attributive adjectives and PredicateAdjective for predicate adjectives. The first one defines two EPTDs corresponding to two different uses of common nouns: heads of noun phrases or predicate complements. Examples (7.11) and (7.12) below illustrate the first cases and Example (7.13) illustrates the second case .

(7.11) *Jean est un ingénieur* **intelligent** *.*
Jean is a engineer clever .
Jean is a clever engineer.

(7.12) *J'ai rencontré une amie et son copain* **espagnols** *.*
I met a friend and her buddy Spanish .
I met a friend and her buddy, who are both Spanish.

(7.13) *Jean est sapeur-pompier* **volontaire** *.*
Jean is firefighter volunteer .
Jean is a volunteer firefighter.

The two EPTDs defined by the RightAttributive class are shown on Figure 7.3. Both have a node *nN* representing the common noun modified by the adjective, except in the case of a coordination of common nouns; in this case, *nN* represents the conjunction of coordination. Therefore, the `cat` feature has the value `n|coord`. Examples (7.11) and (7.12) respectively correspond to *nN* as common noun and coordination conjunction. This node is combined with the adjectival phrase node *nAp* to build its maximal projection *nC*.

Then, the EPTDs differ in the nature of *nC*:

- in the left EPTD, *nC* represents a noun phrase, the noun phrase *un ingénieur intelligent* in Example (7.11) and the noun phrase *une amie et son copain espagnols* in Example (7.12);

- in the right EPTD, *nC* represents an adjectival phrase, the adjectival phrase *sapeur-pompier volontaire* in Example (7.13); as all adjectival phrases, it has a subject represented with node *nSubj2* and this subject co-refers with the subject *nSubj* of the adjectival phrase having the adjective as its head, *volontaire* in our example.

The PredicateAdjective class models the use of adjectives as predicate complements. It inherits the AdjectivalPhrase0 class. The only enrichment with respect to the PTD presented on the right of Figure 7.2 is the addition to node *nAp* of the polarised features `cat` → `ap` and `funct` ← `obj_cpl|mod|obj|objpred|obj_prep|subjpred|void`. The value of `funct` represents the different possible functions of the adjectival phrase.

The RightAttributive and PredicateAdjective classes are gathered in a disjunction AdjectivalPhrase because they concern all adjectives that are able to receive complements, contrary to the LeftAttributive class.

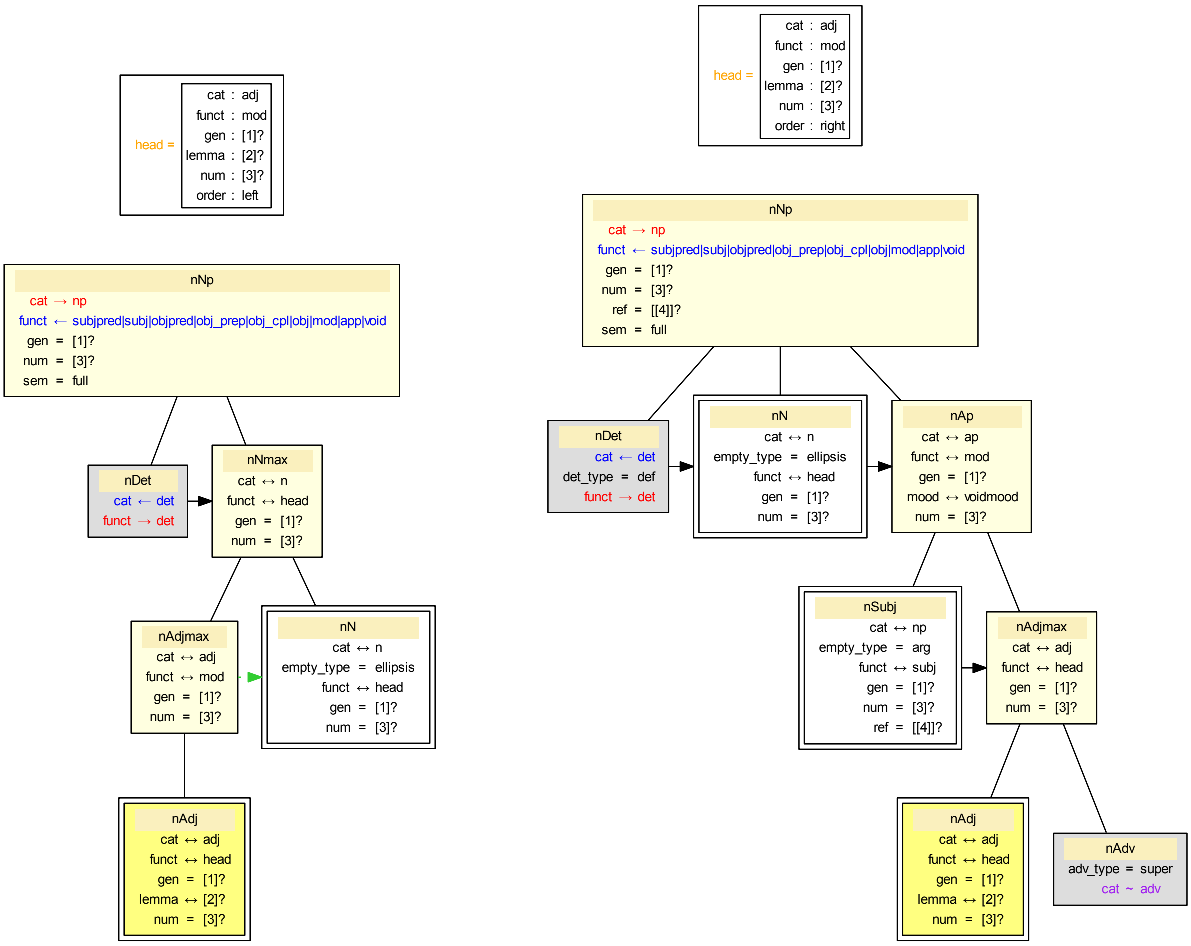

Figure 7.4: PTDs defined by NominalisedLeftAdjective and NominalisedRightAdjective classes

7.2.5 Elision of the nominal head for attributive adjectives

In some contexts, the common noun that is modified by an attributive adjective can be elided, as the following examples show it.

(7.14) *Je connais le père du* **grand** *.*
I knows the father of the tall .
I knows the father of the tall one.

(7.15) *Le plus* **facile** *à faire est de démissionner .*
The easiest to do is to dismiss .

The easiest to do is to dismiss.

Example (7.14) illustrates the ellipsis of the common noun for a left adjective and Example (7.15) illustrates the ellipsis for a right adjective. The PTDs corresponding to the two cases are defined by NominalisedLeftAdjective and NominalisedRightAdjective classes and they are shown in Figure 7.4. The classes respectively inherit the LeftAttributive and RightAttributive classes. Both define an empty node *nN* representing the elided common noun, which is the head of the noun phrase represented with node *nNp*. A node *nDet* represents the expected determiner.

For the right adjective, an additional constraint says that the adjective must be in superlative, which is expressed with node *nAdv*. This constraint is verified in most cases.

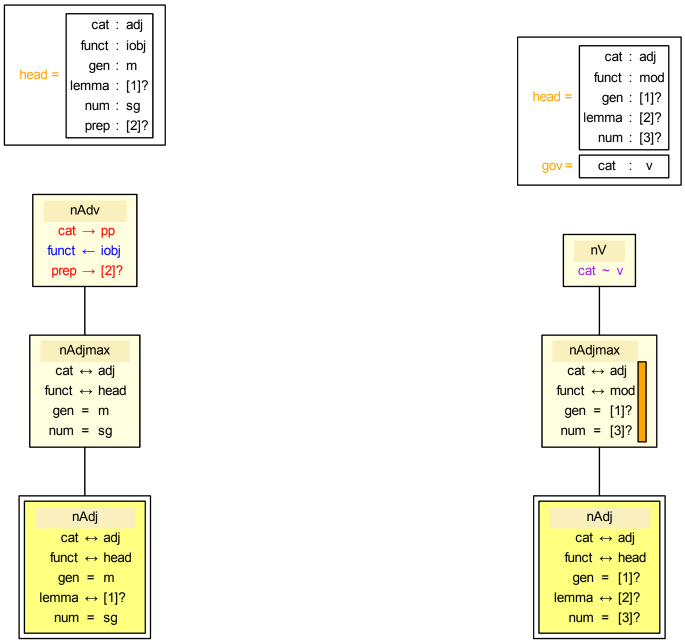

Figure 7.5: EPTDs defined by the ADJAdv class

7.3 Transfer to other categories

Some adjectives in some contexts behave as adverbs or sentences, as the following examples show it.

(7.16) *Jean* *parle* **fort** .
Jean speaks loud .
Jean speaks loud.

(7.17) *Jean* *est* *monté* **haut** *sur* *la* *montagne* *.*
Jean has climbed high on the mountain .
Jean has climbed high on the mountain.

(7.18) **Mince** *!*
Damn !
Damn!

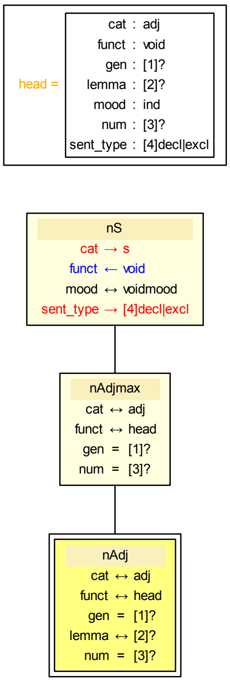

Figure 7.6: EPTD defined by the ADJSENT class

In Examples (7.16) and (7.17), the adjectives behave as adverbs but in the first example, the adverb is considered as a complement required by the verb, whereas in the second example, it is a modifier of the verb. Hence, the ADJADV class defines two EPTDs shown on Figure 7.5. The left EPTD corresponds to the use of the adjective as a required complement. As a consequence, its maximal projection *nAdv* carries three polarized features `cat` → `pp`, `funct` ← `iobj` and `prep` → `?`. In this way, the complement is considered in a uniform way as an indirect object prepositional phrase, like in the sentence *Jean parle d'une voix forte.*

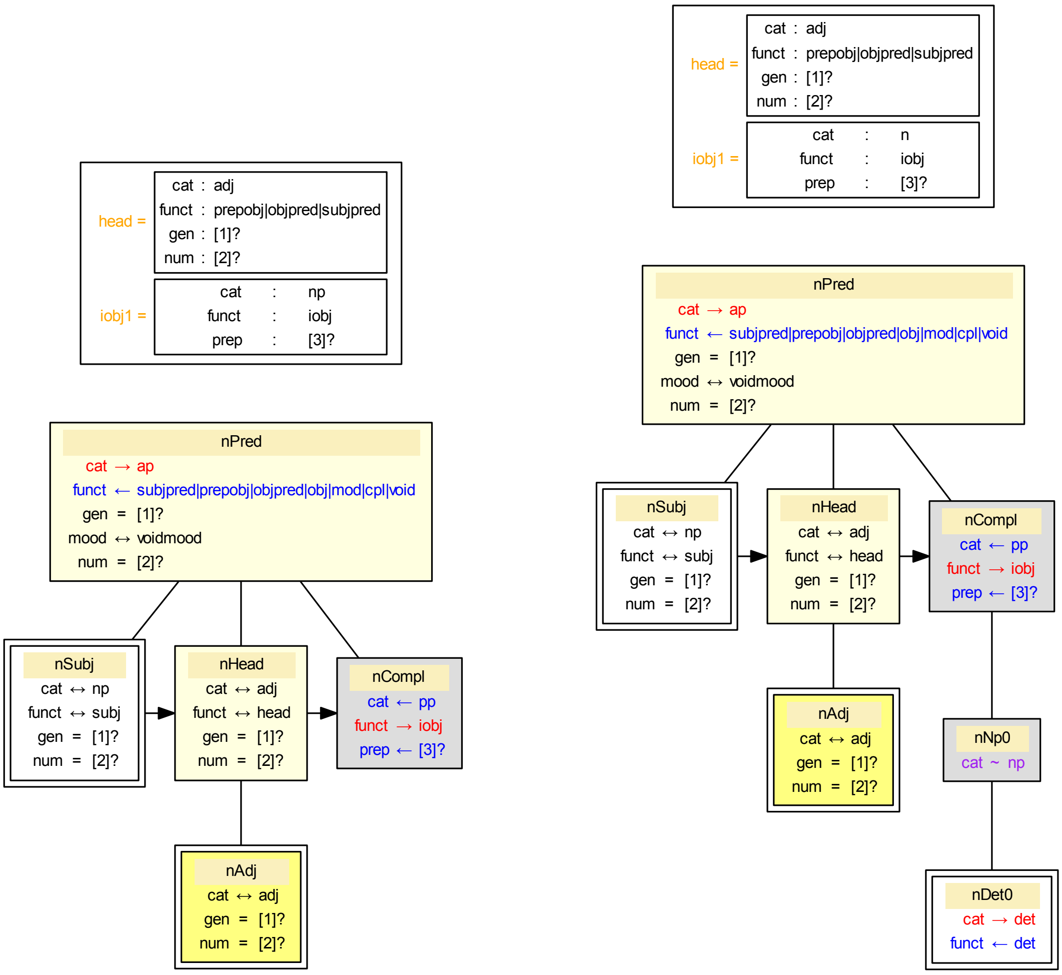

Figure 7.7: EPTDs defined by the ADJ_PP1NOM class

The right EPTD represents a modifier of a verb. Node nV represents the verb after modification by the adjective and the adjective is the rightmost daughter of this node.

Example (7.18) illustrates that some adjectives can be used as the head of exclamative sentences without verb. This phenomenon is modelled with the ADJSENT class and this class generates the EPTD of Figure 7.6. Like verbs, adjectives are sorted according to their valence. Here are examples illustrating the various valences of adjectives.

7.4 Adjectives requiring complements

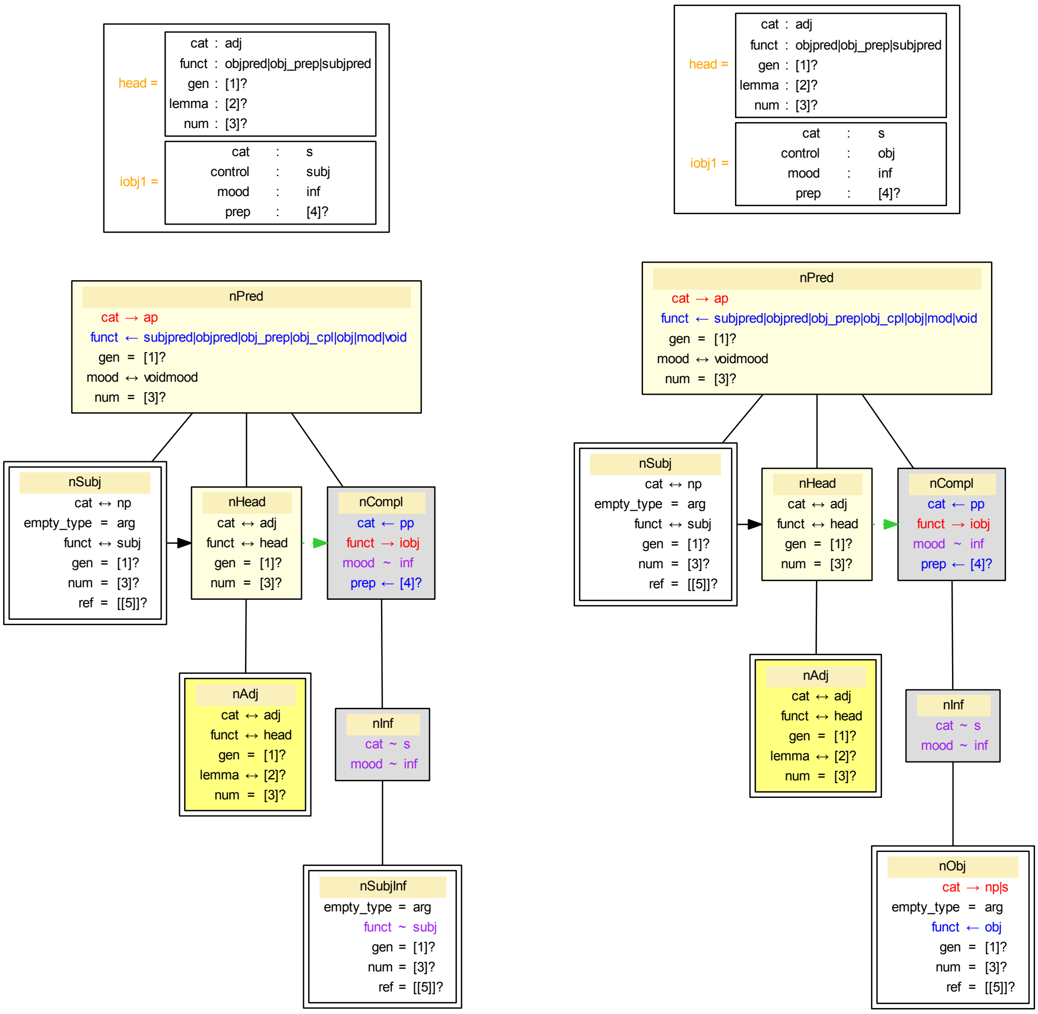

Figure 7.8: EPTDs defined by the ADJ_PP1INF class

(7.19) *Jean* *est* **intelligent** .
Jean is clever .
Jean is clever.

(7.20) *Jean* *est* **doué** *pour* *la cuisine* .
Jean is gifted in cooking .

Jean is gifted in cooking.

(7.21) *Cette situation est* **difficile** *à comprendre .*
That situation is difficult to understand .
That situation is difficult to understand.

(7.22) *Jean est* **lent** *à comprendre .*
Jean is slow-witted in understanding .
Jean is slow-witted in understanding.

(7.23) *Jean est* **jaloux** *que Marie travaille .*
Jean is jealous that Marie works .
Jean is jealous that Marie works.

Adjectives requiring no complement, as in Example (7.19), anchor the EPTDs of the ADJ class, which is the disjunction of four classes: AdjectivalPhrase, LeftAttributive, NominalisedLeftAdjective and NominalisedRightAdjective.

The ADJ_PP1nom class concerns adjectives with a nominal complement, as Example (7.20) illustrates it. It is a disjunction of classes AdjectivalPhrase and NominalisedRightAdjective. Moreover, it inherits the NominalIndirectObject class and it generates 6 EPTDs according to the different functions of the adjective. Figure 7.7 shows the two EPTDs related to the use of adjectival phrases as required complements.

The left EPTD corresponds to adjectives that accepts a noun phrase as a complement, whereas the right EPTD corresponds to adjectives accepting a common noun as a complement. It provides the common noun with an empty determiner represented with node *nDet0*.

The ADJ_PP1inf class concerns adjectives with an infinitive complement, as Examples (7.21) and (7.22) illustrate it. It is a disjunction of classes AdjectivalPhrase and NominalisedRightAdjective. Moreover, it inherits the ClausalIndirectObject class and it generates 6 EPTDs according to the different functions of the adjective. Figure 7.8 shows the two EPTDs related to the use of adjectival phrases as required complements.

In the interface, the `iobj1.control` feature indicates the function of the argument of the infinitive controlled by the subject of the adjective. The left EPTDs concerns adjectives for which the control of the subject of the adjective is over the subject of the infinitive, as in Example (7.22). In the EPTD, the subject of the infinitive is represented with node *nSubjInf*, which co-refers with node *nSubj* representing the subject of the adjective.

The right EPTD concerns adjectives for which the control of the subject of the adjective is over the object of the infinitive, as in Example (7.21). Here, the subject of the adjective *nSubj* co-refers with the object *nObj* of the infinitive.

The ADJ_queS1 class concerns adjectives with a finite clausal complement, as Example (7.23) illustrates it. It inherits the AdjectivalPhrase and FiniteClauseDeObject classes and it generates 3 EPTDs according to the different functions of the adjective. Figure 7.9 shows the EPTD related to the use of adjectival phrases as required complements.

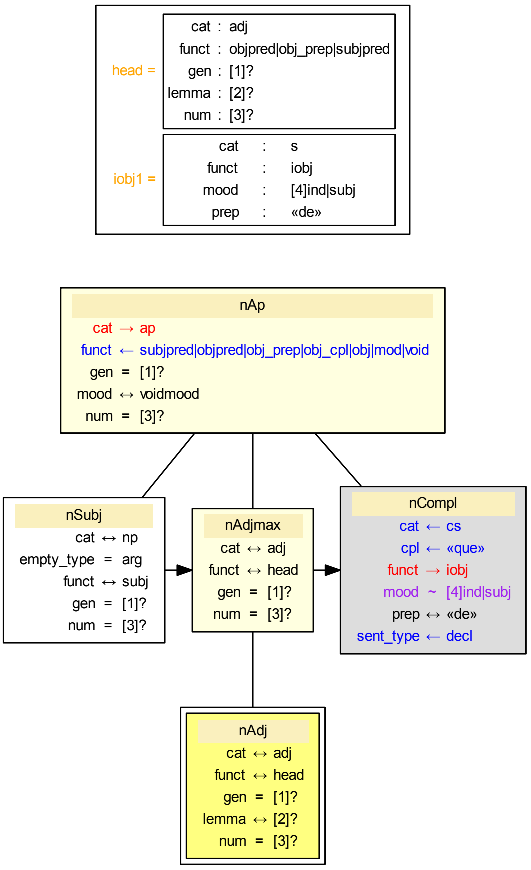

Figure 7.9: EPTD defined by the ADJ_QUES1 class

7.5 Adjectives integrating comparative or consecutive constructions

Some adjectives integrate comparative constructions: *moindre*, *meilleur*, *même*, *autre* ... The adjective *tel* integrate either a comparative construction or a consecutive construction. Here are examples illustrating the use of these adjectives (the concerned adjectives are in bold).

(7.24) *Jean est* **tel** *[que je l'ai toujours connu]* .
Jean is such as I him always knew .
Jean is such as I always knew him.

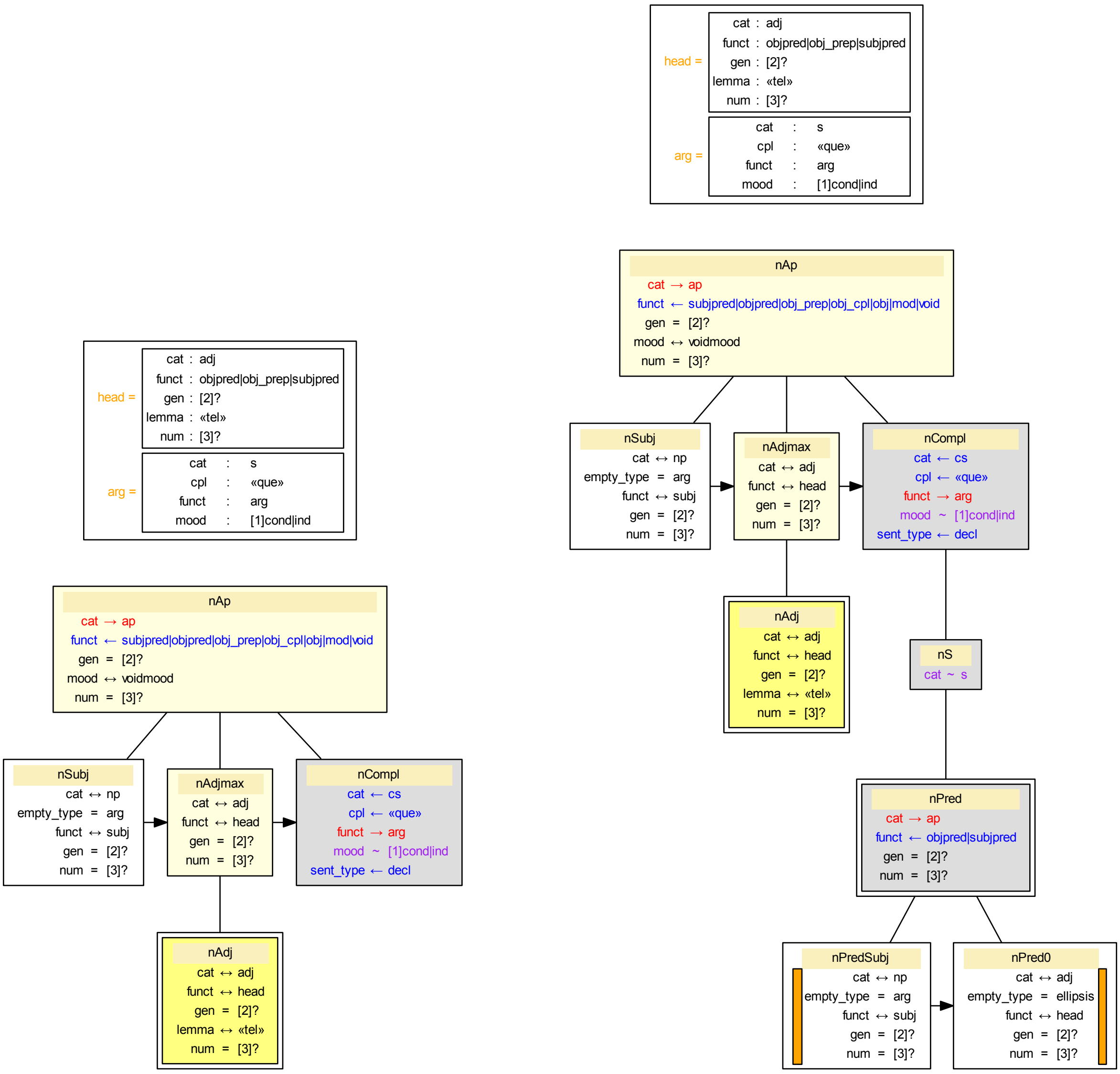

Figure 7.10: EPTDs defined by the ADJ_QUES1COMP-CONS class and used for *tel que* in the parsing of Sentences (7.24) and (7.25)

(7.25)	*La*	*chaleur*	*est*	**telle**	*[que*	*cela*	*devient*	*insupportable]*	.
	The	heat	is	such	that	it	becomes	unbearable	.

The heat is such that it becomes unbearable.

(7.26) *Il connaît la* **même** *personne [que Marie] .*
He knows the same person as Marie .
He knows the same person as Marie.

(7.27) *Un* **autre** *[que lui] aurait présenté les choses différemment .*
Another than him would have presented the things differently .
A person other than him would have presented the things differently.

All these adjectives take a complement clause introduced with the conjunction *que* (between square brackets above). They are not associated with the ADJ_QUES1 class, because they have a particular behaviour, illustrated with the examples above:

- Some of them are left adjectives (Example (7.26)) and the argument clause does not immediately follow the adjective.
- Some elements of the argument clause are elided (Examples (7.24), (7.26) and (7.27)).

The ADJ_QUES1COMP-CONS class defines 10 EPTDs modelling all cases presented in the examples above. Figure 7.10 shows two EPTDs corresponding to the two first examples.

The left EPTD corresponds to Example (7.25). Node *nCompl* represents the clause argument of the adjective. A positive feature `funct` → `arg` expresses that the adjective will provide the expected complemented clause with the function `arg`.

The right EPTD corresponds to Example (7.24). It is more complicated because in the complement clause represented with node *nCompl*, there is an adjective which is elided. It is represented by the empty node *nPred0.* As all adjectives, it has an empty subject represented with node *nPredSubj.* Together they build the noded *nPred*, which is labeled with the negative feature `funct` ← `objpred|subjpred`: it indicates that it can receive the function of predicate complement. In Example (7.24), it is an object predicate complement of the verb *connu.*

Chapter 8

Adverbs

Adverbs constitute a residual class which is not strictly delimited. In a first approach, they are invariable words used to modify various types of constituents: sentences, noun phrases, prepositional phrases, verbs, adjectives . . . and even adverbs.

The classes anchored with adverbs are gathered in the Adverb module.

8.1 Interfaces with the lexicon

Adverbs are characterised in the interface with the feature `head.cat = adv`. Their properties are described with the following sub-features of the `head` feature:

- `adv_type`: it indicates the type of the adverb, `adj` (adverb used as an adjective), `inter` (interrogative), `neg` (the particle *ne*), `negcompl` (negative adverb used in conjunction with the particle *ne*), `stand` (standard);
- `funct`: it describes the possible syntactic functions among `obj_cpl`, `iobj`, `mod`, `obj`, `objpred`, `obj_prep`, `subj`, `subjpred`;
- `prep`: when the adverb has the same function as a prepositional complement (locative complement for instance), it indicates an implicit preposition;
- `order`: it concerns the position of the adverb when it acts as a modifier in the modified constituent; it can take the values `left`, `right` or `neutr` if it respectively si before the head, after the head or in any position;
- `sent_type`: when an adverb is the head of a sentence, it gives the type of the sentence, `decl` (declarative), `excl` (exclamatory), `imper` (imperative) or `inter` (interrogative).

When an adverb is used as a modifier, a feature `gov.cat` gives the category of the constituent that is modified: `adj` (adjective), `ap` (adjectival phrase), `adv` (adverb), `cs` (complemented sentence), `np` (noun phrase), `pp` (prepositional phrase), `ps` (clause introduced by a preposition), `s` (sentence), `v` (verb), `vp` (verb phrase).

Some adverbs take a clausal complement introduced with the preposition *de* and the properties of this complement are described with the following sub-features of a feature `iobj1`:

- `cat`: it indicates the category of the complement, `s` usually;
- `cpl`: it indicates the complementizer introducing the complement clause, `que` or `voidcpl` (non complementizer);
- `funct`: it indicates the function of the complement, `iobj` usually;
- `mood`: it indicates the mood of the complement clause, `ind` (indicative), `inf` (infinitive) or `subj` (subjunctive);
- `prep`: it indicates the preposition introducing the complement, `de` usually.

For adverbs that are correlated with conjunctions or prepositions (*plus*. . .*que*, *trop*. . .*pour*, . . .), the interfaces include a special feature `arg` to describe the properties of the clause introduced by these conjunctions or prepositions. This feature uses the same sub-features as feature `iobj1`.

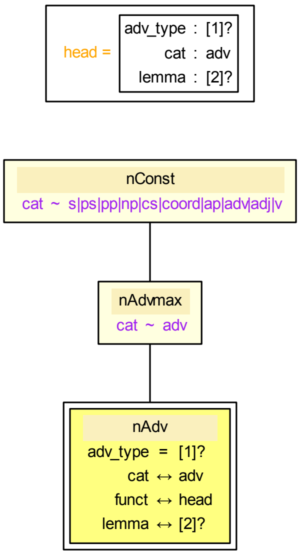

Figure 8.1: The PTD defined by the Adverb class

Since interrogative adverbs have a very specific behaviour, which make them closer to other interrogative words, they are not described in this chapter but in chapter 10.

8.2 The different functions of adverbs

A basic class Adverb defines the common skeleton of all EPTDs anchored by adverbs and shown on Figure 8.1. Node *nAdvmax* represents the kernel headed by the adverb anchored at node *nAdv*. The mother node *nConst* of *nAdvmax* must appear in the PTD because its category depends on the adverb.

8.2.1 Adverbs as indirect objects of verbs

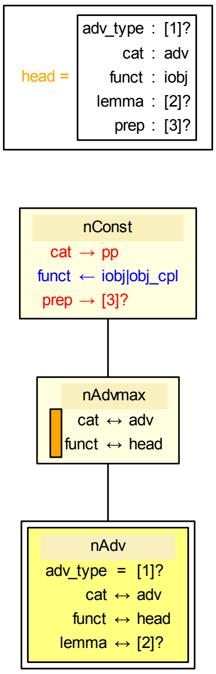

Figure 8.2: The EPTD defined by the IndirectObjectAdverb class

Some adverbs can play the role of indirect objects of verbs as the following examples illustrate it.

(8.1) *Jean va* **là-bas** *plutôt qu'* **ailleurs** .
Jean goes there rather than elsewhere .
Jean goes there rather than elsewhere.

(8.2) *Jean va* **mieux** *que son frère* .
Jean seems better than his brother .
Jean seems to feel better than his brother.

The IndirectObjectAdverb class defines one EPTD for the adverbs used as indirect object, which is shown in Figure 8.2. It inherits the Adverb class and it adds polarised features to node *nConst*, which is the maximal projection of the adverb. This node represents a prepositional phrase because in this case the adverb is considered to play same role as a complement prepositional phrase. Hence, it carries a positive feature `prep` $\rightarrow$ `?`, which will take the value `loc` in Example (8.1) and `voidprep` in Example (8.2). The function that is expected by *nConst* is usually `iobj` but sometimes it may take the function `cpl` when the adverb depends on the conjunction in a comparison as *ailleurs* in Example (8.1).

8.2.2 Adverbs as noun phrases

Some quantitative adverbs can behave as noun phrases.

(8.3) *Jean mange* **beaucoup** *.*
Jean eats very much .
Jean eats very much.

(8.4) *Jean mange* **moins** *que Marie .*
Jean eats less than Marie .
Jean eats less than Marie.

(8.5) *Jean vient de* **loin** *.*
Jean comes from afar .
Jean comes from afar.

(8.6) *Jean connaît* **plus** *d'histoires que Marie .*
Jean knows more stories than Marie .
Jean knows more stories than Marie.

The NounPhraseAdverb class expresses the behaviour of these adverbs. It inherits the Adverb class and adds polarised features as IndirectObjectAdverb. Here, the polarised features are `cat` $\rightarrow$ `np` and `funct` $\leftarrow$ `obj_cpl|obj|objpred|obj_prep|subj`.

For quantitative adverbs that they have a partitive complement introduced with *de*, like in Example (8.6), the NounPhraseAdverb class is specialised in the NounPhraseAdverbWithComplement class. It defines the EPTD presented on Figure 8.3. The partitive complement is represented with node *nCompl*. This node has a feature `det_type`, which share its value with the feature `iobj1.det_type` of the interface because the determiner of the complement depends on the adverb. For instance, for *trop* the value is `voiddet`, whereas for *plus* it is `indef|voiddet`.

In all examples above, the function of the adverb is `obj` except Example (8.5) where the adverb *loin* has the function `obj_prep`.

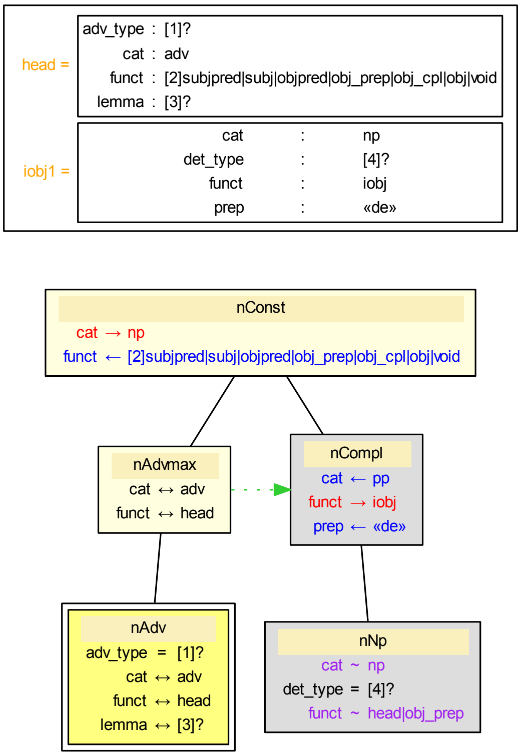

Figure 8.3: The EPTD defined by the NounPhraseAdverbWithComplement class

8.2.3 Adverbs as sentence heads

Some adverbs, like *alors*, *bien*, *oui*, *tant pis* . . ., behave as sentence heads. The ADVSent class models this phenomenon and it defines three EPTDs according to the type of the sentence.

Figure 8.4 shows the three EPTDs. From the left to the right, they correspond to declarative or exclamatory sentences, interrogative sentences or imperative sentences. The difference between them lies in the value of the feature `sent_type` which is carried by the node *nS* representing the sentence. This value is shared by the feature `head.sent_type` of the interface.

The last and most usual function of adverbs is modifier, which is addressed in the next section.

The IndirectObjectAdverb, NounPhraseAdverb, NounPhraseAdverbWithComplement classes are used as intermediate classes for negative, comparative and other adverbs but they are also used as terminal classes for standard adverbs. In this case, they are gathered in a disjunction ADVArg.

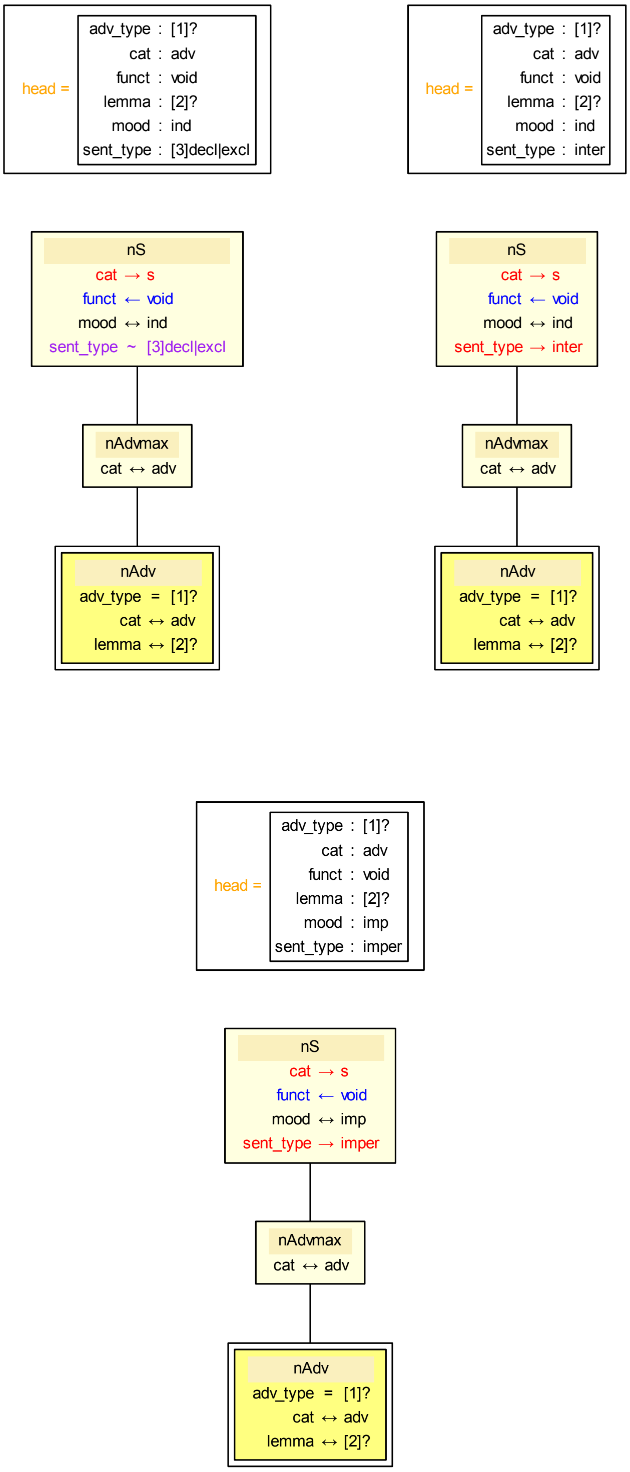

Figure 8.4: The EPTDs defined by the ADVSENT class

8.2.4 The specific case of *que*

The adverb *que* can be used as head of a sentence with ellipsis to mark its exclamative feature, but this marking can occur in another context. Here are two sentences illustrating these two uses of *que*.

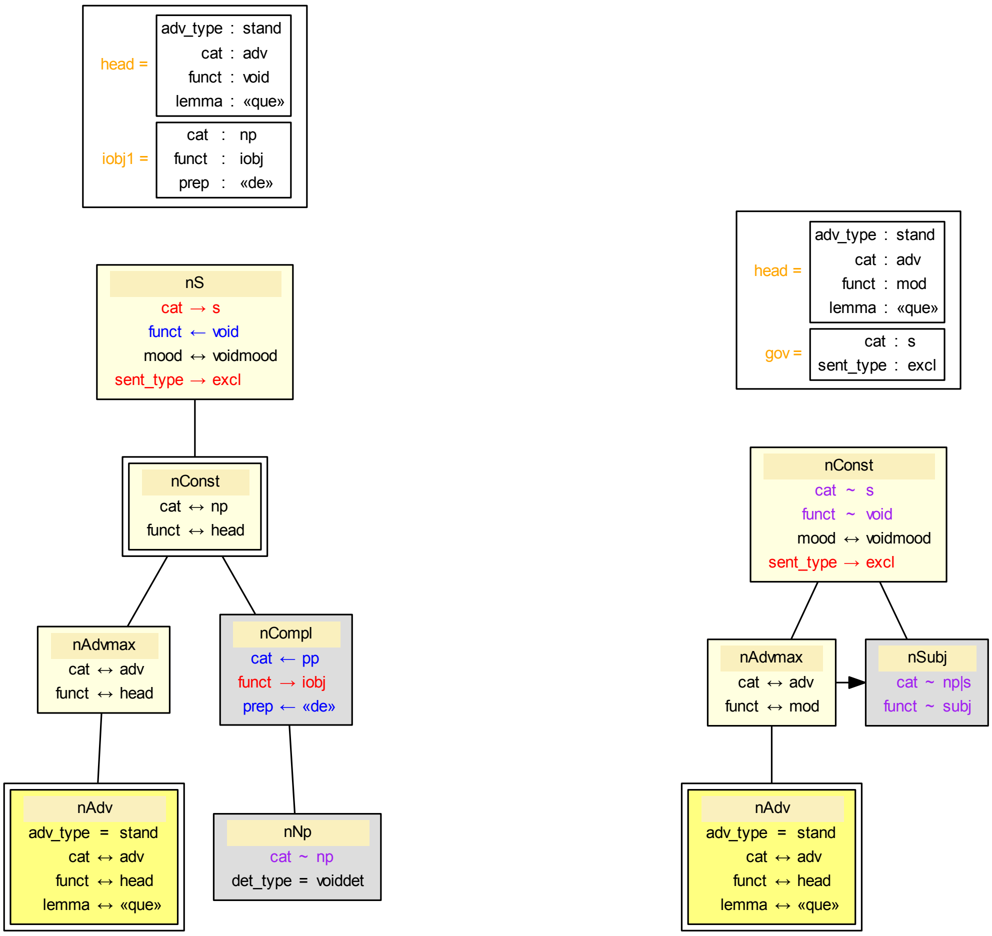

Figure 8.5: The EPTDs defined by the ADVQUE-EXCL class

(8.7) *Que d' eau !*
What of water !
So much water!

(8.8) *Aujourd'hui, que l' eau est froide !*
Today, what the water is cold !
Today, the water is so cold!

The ADVQue-excl class deals with these cases. It inherits the Adverb class and defines the two EPTDs shown in Figure 8.5. The EPTD on the left is used in Example (8.7). The adverb *que* is the head of a noun phrase represented by Node *nConst* and it requires a complement introduced with *de*.

The EPTD on the right is used in Example (8.8). It considers the adverb as sentence modifier, with the constraint that *que* precedes the subject of the sentence immediately.

8.3 Adverbs as modifiers

Most often, adverbs are not complements required by verbs but they modify different kinds of words or phrases. The ModifierAdverb class defines the common skeleton of all EPTDs expressing this function, which is shown on Figure 8.6.

It inherits the Adverb class and renames node *nConst* as *nModif* because this node represents the constituent modified by the adverb. The value of its feature `cat` gives the categories of constituents possibly modified by adverbs.

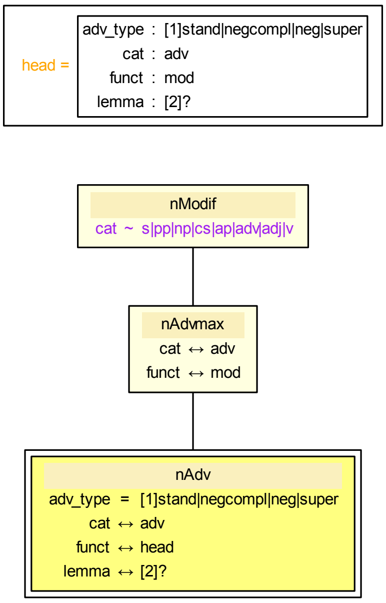

Figure 8.6: The EPTD defined by the ModifierAdverb class

Then, according to the grammatical category modified by the adverbs, the ModifierAdverb class is specialised in various classes.

8.3.1 Adverbs as sentence modifiers

A criterion for detecting sentence modifiers, is that they can be put at the beginning of a sentence, but their position may be relatively free, as the following examples show it.

(8.9) *Jean voit Marie* **aujourd'hui** *.*
Jean sees Marie today .
Jean sees Marie today.

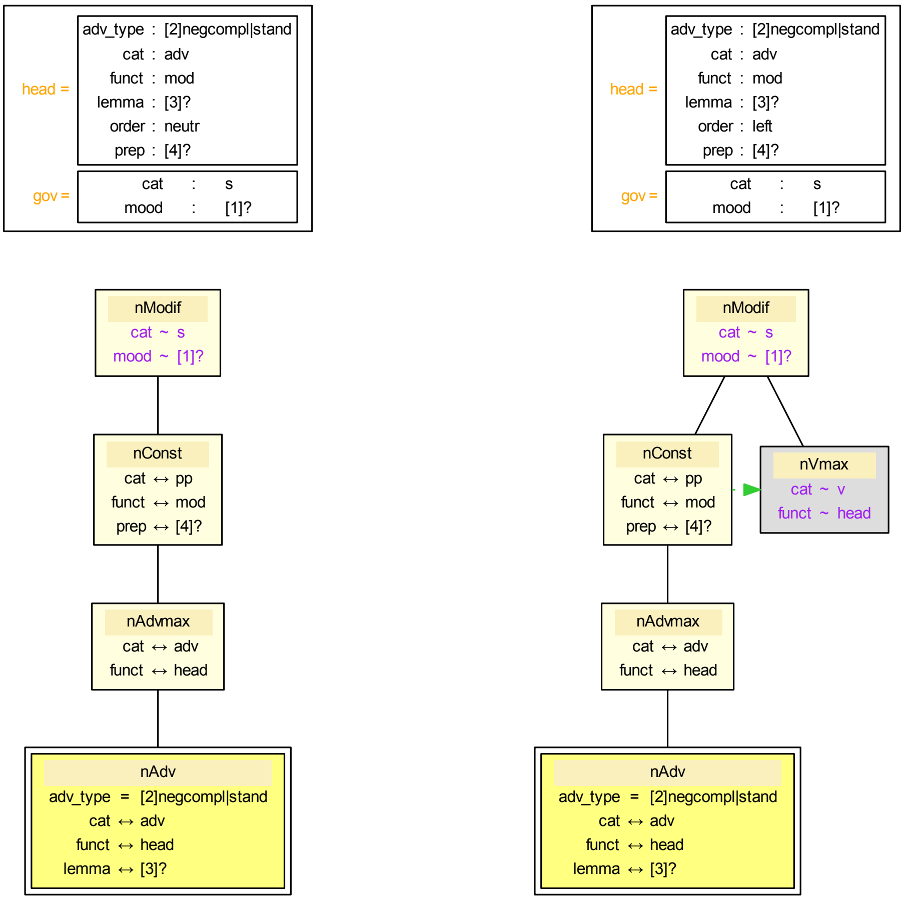

Figure 8.7: The EPTDs defined by the SentenceModifierAdverb class

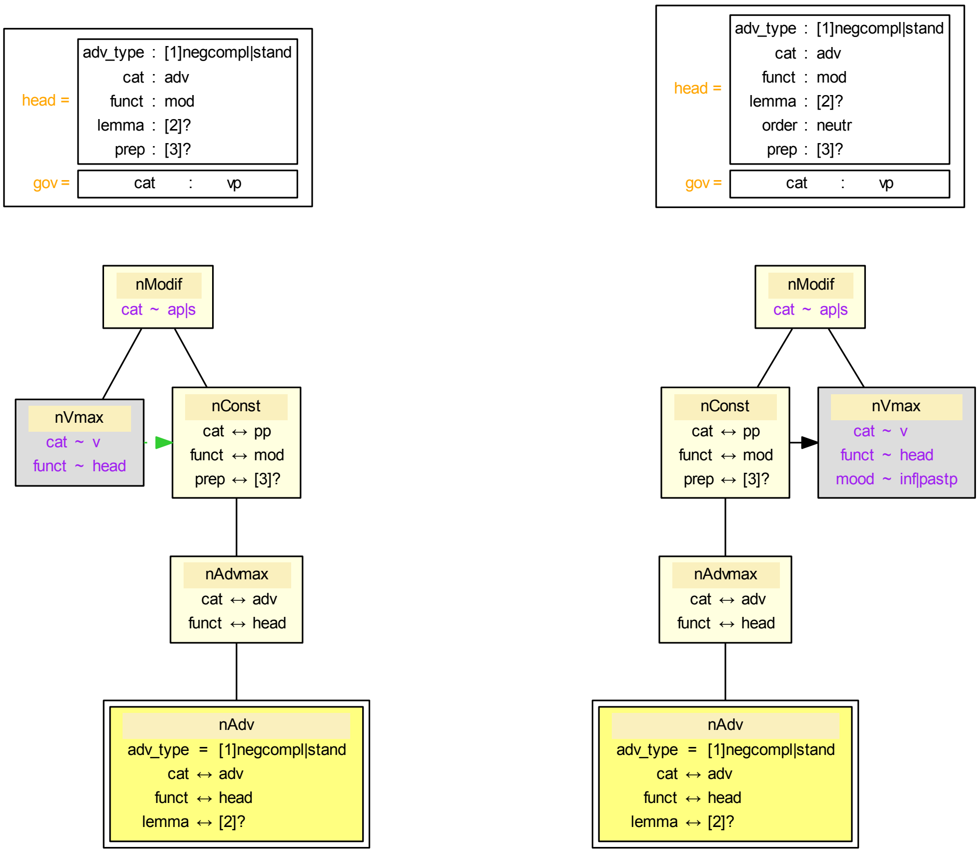

Figure 8.8: The EPTDs defined by the VERBPHRASEMODIFIERADVERB class

(8.10) **Aujourd'hui**, *Jean voit Marie .*
Today, Jean sees Marie .
Today, Jean sees Marie.

(8.11) *Marie indisponible* **aujourd'hui** *viendra demain .*
Marie unavailable today will come tomorrow .
Marie today unavailable will come tomorrow

(8.12) *Marie* **aujourd'hui** *indisponible viendra demain .*
Marie today unavailable will come tomorrow .
Marie today unavailable will come tomorrow.

(8.13) **Jamais** *Jean ne voit Marie .*
Never Jean sees Marie .
Jean never sees Marie.

(8.14) **Jean ne voit Marie* **jamais** *.*
Jean sees Marie never .
Jean never sees Marie.

Some adverbs modifiers of sentences, like *aujourd'hui*, have a totally free position in the sentence, but other ones, like *jamais*, must be put before the verb[1].

Hence, we have two corresponding EPTDs defined by the SentenceModifierAdverb class and presented on Figure 8.7. The left EPTD corresponds to the case that the position of the adverb is free in the sentence, which is indicated in the interface with the feature `order = neutr`. The right EPTD corresponds to the case of the adverb that must precede the verb, which is indicated in the interface with the feature `order = left`.

8.3.2 Adverbs as verb phrase modifiers

Since our grammar ignores verb phrases, we cannot represent adverbs modifying verb phrases explicitly. We represent them as sentence modifiers too but constraining the adverb to occur after the verb it depends, except if the verb is an infinitive or a past participle.

(8.15) *Jean démonte son ordinateur* **complètement** *.*
Jean dismantles his computer completely .
Jean dismantles his computer completely.

(8.16) *Jean a démonté* **complètement** *son ordinateur .*
Jean has dismantled completely his computer .
Jean has completely dismantled his computer.

(8.17) *Jean a* **complètement** *démonté son ordinateur .*
Jean has completely dismantled his computer .
Jean has completely dismantled his computer.

(8.18) ***Complètement** *Jean a démonté son ordinateur .*
Completely Jean has dismantled his computer .
Jean has completely dismantled his computer.

The VerbPhraseModifierAdverb class defines 2 EPTDs shown in Figure 8.8. The general case, illustrated with Sentences (8.15) and (8.16), corresponds to the left EPTD and the particular case for infinitives and past participles, illustrated with Sentence (8.17), corresponds to the right EPTD.

8.3.3 Adverbs as modifiers of other categories

For other categories of phrases and words, there are other classes taking their specifies into account (the category of the modified phrase or word, the position of the adverb...):

[1] As verb modifier, *jamais* can be put in the verb kernel as in the sentence *Jean ne voit jamais Marie.*

- the ComplementedClauseModifierAdverb class for adverbs modifying complemented clauses: they are always put at the beginning of the complemented clause;
- the PrepositionalPhraseModifierAdverb class for adverbs modifying prepositional phrases; they are always put at the beginning of the prepositional phrase;
- the VerbModifierAdverb class for adverbs modifying verbs; generally, their position with respect to the modified verb depends on the mood of the verb but for particular adverbs, they are always put immediately after the verb;
- the AdjectiveAdverbModifierAdverb class for adverbs modifying adjectives or adverbs; they are always put at the beginning of the modified constituent;
- the AdjectivalPhraseModifierAdverb class for adverbs modifying adjectival phrases; we distinguish them from adverbs modifying adjectives; the position of the first ones is relatively free in the adjectival phrase, whereas the second ones are put before adjectives;
- the LeftNounPhraseModifierAdverb and RightNounPhraseModifierAdverb classes for adverbs modifying noun phrases; as for adverbs modifying adjectives, we distinguish two cases according to the position of the adverb with respect to the noun phrase; when they are put at the end, they are assimilated to prepositional phrases.

All classes of this section concern the following types of adverbs: `stand` and `negcompl`. For the first one, the classes are gathered in a disjunction ADVmodif_C1, which is a terminal class but for the second one, the classes must be enriched as the next section will describe it.

8.3.4 Superlatives

The expressions *le plus* and *le moins* are used to express superlatives, as the following examples show it:

(8.19)	*Cette*	*histoire*	*est*	**la**	**plus**	*facile*	*à*	*comprendre*	*.*
	This	story	is	the	most	easy	to	understand	.

This story is the easiest to understand.

(8.20)	*Marie*	*est*	*ma*	**moins**	*grande*	*sœur*	*.*
	Marie	is	my	less	tall	sister	.

Marie is my least tall sister.

(8.21)	*Marie*	*est*	*ma*	*soeur*	**la**	**moins**	*grande*	*.*
	Marie	is	my	sister	the	less	tall	.

Marie is my least tall sister.

When the adjective is attributive and precedes the noun that it modifies, as in Sentence (8.20), the construction is considered as a standard modification of an adjective. In our example, *moins* is a modifier of *grande*. The superlative feature of the construction is entailed by the nature of the determiner, which must be definite or possessive.

In other constructions of the adjective, the expressions *le plus*, *le moins*, with their flexions, are considered as compound adverbs, even if they are not invariable. In this case, they always enter superlative constructions and they anchor a specific EPTD defined by the ADVsuper_C1 class. Figure 8.9 shows this EPTD.

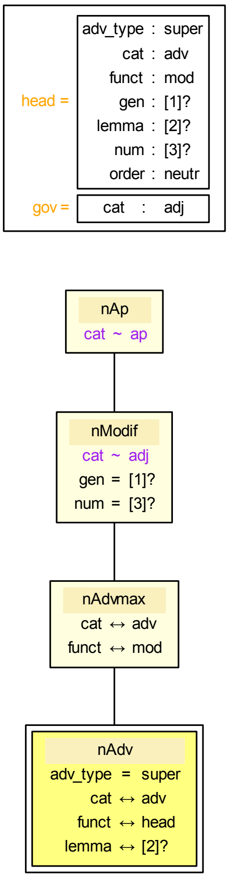

Figure 8.9: The EPTD defined by the ADVsuper_C1 class

8.4 Negation adverbs

In French, negation is most often expressed with the particle *ne* paired with a satellite word, which can be an adverb, a pronoun or a determiner: *pas*, *personne*, *aucun* ... The following examples illustrate the case of adverbs as satellites of the particle *ne*.

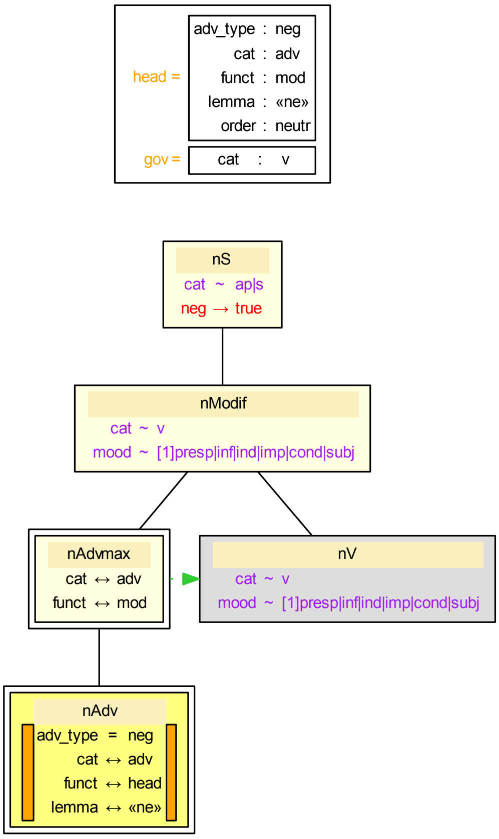

Figure 8.10: The EPTD defined by the ADVNE_V1 class

(8.22) *Jean* **ne** *mange* **que** *des* *pommes* .
Jean eats only apples .
Jean eats only apples.

(8.23) *Jean* **ne** *pense* *pouvoir* *travailler* **que** *dans* *sa* *chambre* .
Jean thinks to be able to work only in his room .
Jean thinks to be able to work only in his room.

(8.24) *Jean* **ne** *mange* **pas** **que** *des* *pommes* .
Jean does not eat only apples .
Jean does not eat only apples.

This pairing is expressed in FRIGRAM with a polarized feature `neg`, which is attached at the clause constituting the scope of the negation.

The *ne* particle provides the positive feature `neg` → `true` to neutralize the dual negative feature `neg` ← `true` given by the satellite negative word.

Hence, the ADVne_V1 class defines the EPTD presented on Figure 8.10 for the the *ne* particle, taken as an adverb. Node *nS* represents the sentence that is the scope of the negation. It carries the positive feature `neg` → `true`. The particle *ne* appears as a clitic put before the verb represented by the node nV. The maximal projection of the clitic represented by the node *nAdvmax* cannot receive any modifiers, which is indicated by the fact that the node is closed (double rectangle on the figure).

The satellite word paired with the *ne* particle can be a pronoun or a determiner but here, we only consider adverbs as *pas*, *point*, *plus*, *guère*, *jamais*, *nulle part* or *que*. A satellite adverb follows the syntax of any adverb described in the previous section: it can be an argument or a modifier.

A difficulty comes from the fact that one particle *ne* can be paired with several satellites, as Example (8.24) shows it. In this sentence, there are two satellite adverbs: *pas* and *que*. Now, in the IG formalism, one positive feature `neg` → `true` must be saturated by exactly one negative feature `neg` ← `true`.

Our solution is to distinguish between a main satellite word and a secondary satellite word. The main satellite word brings the negative feature `neg` ← `true` and the secondary satellite word brings a virtual feature `neg` ∼ `true`.

A unique class SatelliteNegationAdverb takes both cases into account. Moreover, since a satellite negation adverb behaves like other adverbs, it inherits different classes in a disjunctive way: IndirectObjectAdverb, SentenceModifierAdverb, ComplementedClauseModifierAdverb, PrepositionalPhraseModifierAdverb, VerbModifierAdverb, AdjectiveAdverbModifierAdverb and LeftNounPhraseModifierAdverb.

In the disjunction, two specific properties are added for *que*, when it modifies an infinitive or an adjectival phrase, because it is put at the beginning of the constituent that it modifies.

Figure 8.11 shows two of the 24 PTDs defined by the SatelliteNegationAdverb class, for a negation adverb that is a modifier of a noun phrase:

- The left one corresponds to a main satellite negation adverb. Node *nS* represents the clause or the adjectival phrase that is the scope of the negation: its head is cliticised with the particle *ne*. Node *nS0* represents the clause or the adjectival phrase that contains the satellite negation adverb. There is an underspecified dominance relation from *nS* to *nS0* because *nS0* can identify with *nS* or it can be embedded in *nS*. Since the adverb is a main satellite negation adverb, it adds the negative feature `neg` ← `true` to the node *nS*, which will be neutralised by the dual positive feature brought by the particle *ne*.

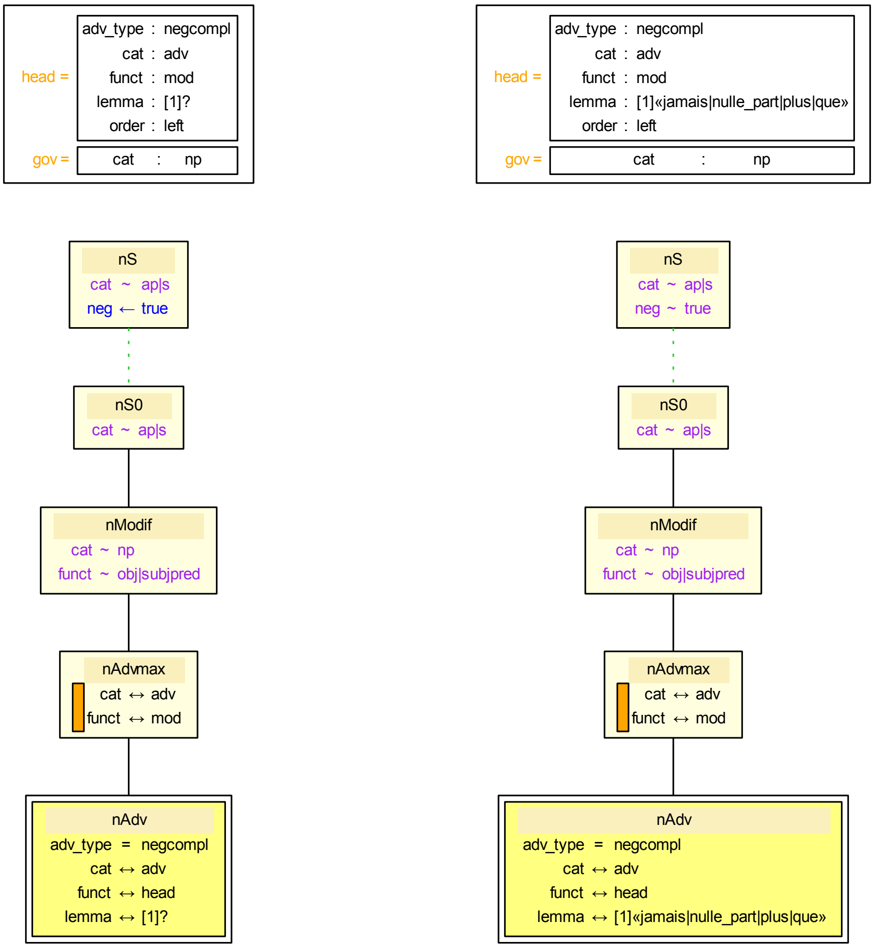

Figure 8.11: PTDs defined by the SatelliteNegationAdverb class for negation adverbs modifying noun phrases

Example (8.22) illustrates the use of this PTD. In this sentence, *nS* and *nS0* identify and represents the whole sentence. Node *nModif* represents the noun phrase *que des pommes*, modified by the adverb *que* represented by node *nAdvmax*.

- The right one corresponds to a secondary satellite negation adverb. The only difference with respect to the previous PTD is the polarity of feature `neg`, which is virtual. Example (8.24) illustrates the use of this PTD. The PTD is attached at the secondary satellite adverb *que* and the main satellite adverb is *pas*.

The possibility of dissociating *nS0* from *nS* comes from the observation that a satellite adverb may be embedded more or less deeply in an object infinitive clause, like in Sentence (8.23). Thus, the SatelliteNegationAdverb class is divided into two subclasses corresponding to two cases :

- the satellite adverb is in the same clause as the *ne* particle, which is expressed with the DirectSatelliteNegationAdverb class; Sentences (8.22) and (8.24) illustrate this case;

- the satellite adverb is embedded more or less deeply in an object infinitive clause inside the clause containing the *ne* particle, which is expressed with the IndirectSatelliteNegationAdverb class; Sentence (8.23) illustrates this case.

Figure 8.12 presents an example of PTD defined by each class. The left one is associated with the *que* adverb to parse Sentence (8.22) and it is defined by the class DirectSatelliteNegationAdverb. Node *nModif* represents the noun phrase *que des pommes*, which is an immediate sub-constituent of the sentence *Jean ne mange que des pommes*, the scope of the negation, represented with node *nS0*.

The right one is associated with the *que* adverb to parse Sentence (8.23) and it is defined by the IndirectSatelliteNegationAdverb class. Node *nS* represents the scope of the negation, the whole sentence. Node *nS1* represents the infinitive which an immediate sub-constituent, the clause *pouvoir travailler que dans sa chambre*. Node *nS0* represents the infinitive *travailler que dans sa chambre* and node *nModif* represents the modified prepositional phrase *que dans sa chambre*. The domination of node *nS1* over node *nS1* is underspecified, because any number of infinitives can be embedded between them.

The DirectSatelliteNegationAdverb and IndirectSatelliteNegationAdverb classes are gathered in their disjunction, ADVnegcompl_C1.

Some adverbs (*guère*, *jamais*, *non*, *nulle part*, *pas*) can be used alone when they modify an attributive adjectival phrase, like in the following sentences.

(8.25) *On* *lui* *donne* *un* *travail* **pas** *facile* *.*
One him gives a work not easy .
One gives him a not easy work.

(8.26) *Cela* *demande* *un* *travail* **non** *négligeable* *.*
It requires a work non insignificant .
It requires a non insignificant work.

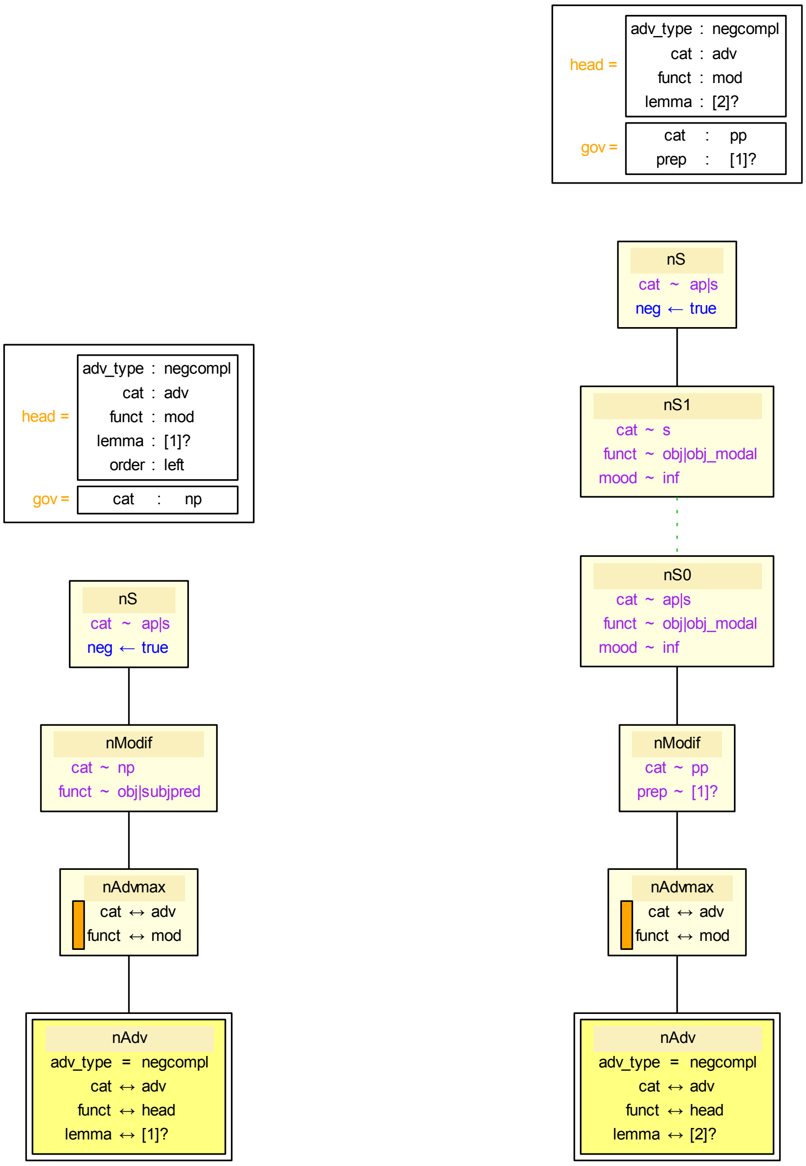

Figure 8.12: EPTDs defined by the DirectSatelliteNegationAdverb and IndirectSatelliteNegationAdverb classes respectively used in Example (8.22) and 8.23

(8.27) *C' est un sport pratiqué* **nulle part** *.*
That is a sport practised nowhere .
That is a sport nowhere practised.

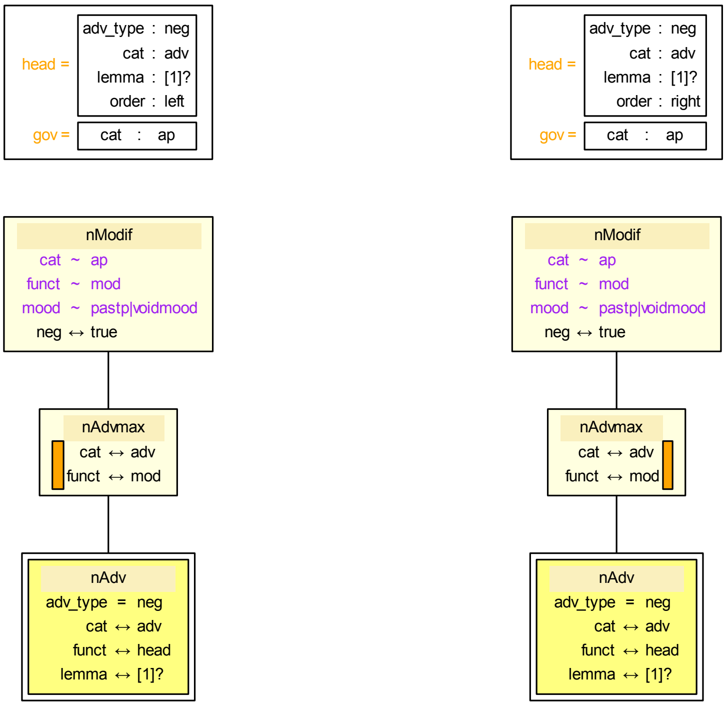

Figure 8.13: The EPTDs defined by the ADVNEG_AP1 class

This specific use is defined by the ADVNEG_AP1 class, which produces the EPTDs of Figure 8.13. Both correspond to negation adverbs acting as modifiers of attributive adjectival phrases. The left EPTD corresponds to left modifiers (Examples (8.25) and (8.26)) and the right EPTD corresponds to right modifiers (Example (8.27)).

8.5 Adverbs used as adjectives

Some adverbs can be used as adjectives as the following examples illustrate it.

(8.28) *Jean est* **bien** *.*
Jean is okay .
Jean is okay.

(8.29) *Jean la trouve* **mieux** *que son frère .*
Jean her finds better than her brother .
Jean finds her better than her brother.

(8.30) *Jean est un type* **bien** *.*
Jean is a guy good .

Jean is a good guy.

(8.31) *J'ai* *vu* *Jean* **en train** *de* *réparer* *sa* *voiture* *.*
I saw Jean being repairing his car .
I saw Jean repairing his car.

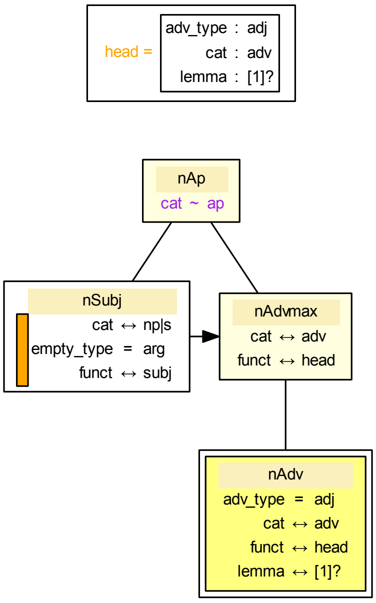

Figure 8.14: The PTD defined by the AdjectiveAdverb class

The AdjectiveAdverb class defines the skeleton of all EPTDs anchored by adverbs behaving as adjectives. It is shown on Figure 8.14. Node *nAp* represents an adjectival phrase having the adverb as its head. As any adjectival phrase, it has empty subject represented with node *nSubj*.

Then, the class is specialised in two sub-classes, according to the function of the adverb:

- the PredicateAdverb class defines the EPTD for the adverbs used as predicate adjectives, as it is illustrated with Sentences (8.28), (8.29) and (8.31); this EPTD is shown on the left part of Figure 8.15;
- the AttributiveAdverb class defines the EPTD for the adverbs used as attributive adjectives, as it is illustrated with Sentence (8.30); this EPTD is shown on the right part of Figure 8.15;

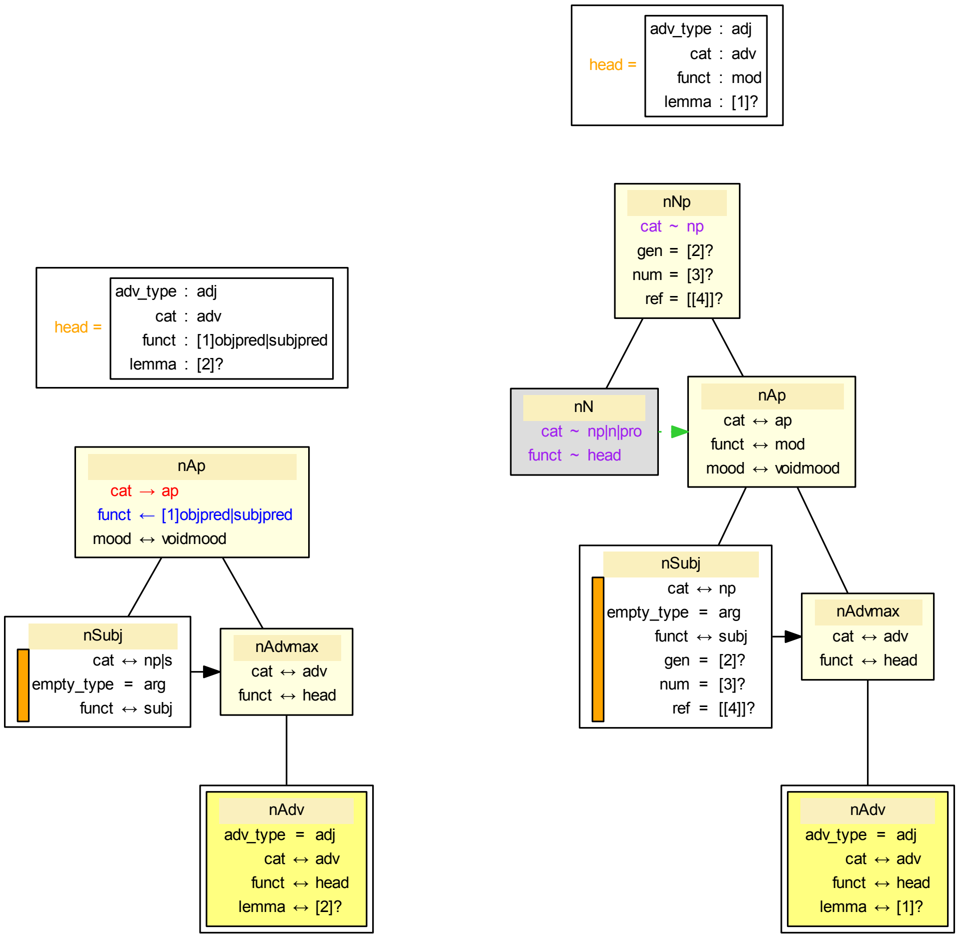

Figure 8.15: The EPTDs defined by the PredicateAdverb and AttributiveAdverb classes

The PredicateAdverb and AttributiveAdverb classes are gathered in the disjunction ADVAdj, which is a terminal class.

As Example (8.31) shows it, adverbs with the function of adjectives can take infinitive complements. The ADVAdj_deS1inf class models this case and it defines two EPTDs. Figure 8.16 shows the EPTD corresponding to the predicate function of the adverb, illustrated by our example.

8.6 Adverbs correlated with complement clauses

Some adverbs governs a complement clause in comparative or consecutive constructions, as the following examples show it.

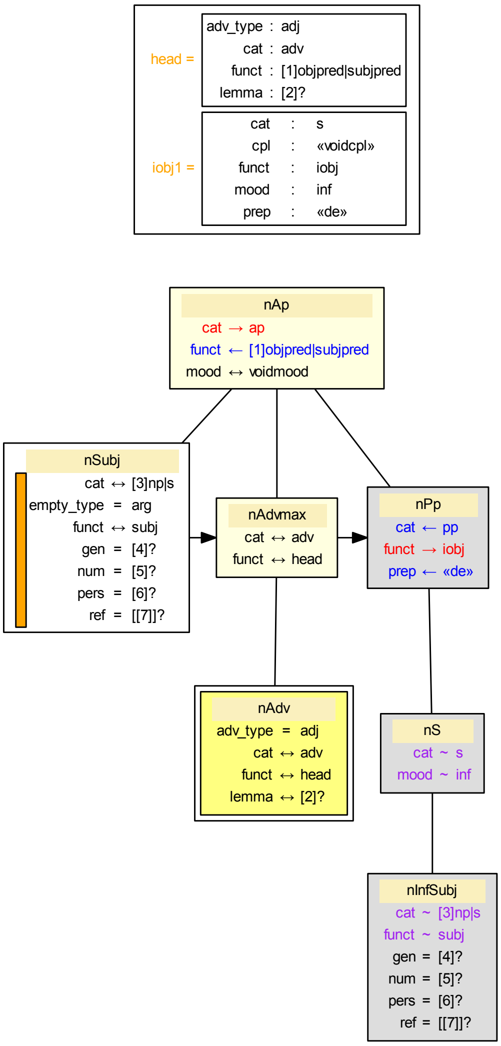

Figure 8.16: The EPTD defined by the AdvAdj_deS1inf class for the predicate function of the adverb

(8.32) *Jean* *connaît* *les* *parents* *de* **trop** *d'élèves* **pour** *ne pas* *venir* *.*
Jean knows the parents of too many students to not come .
Jean knows the parents of too many persons to not come.

(8.33) *Jean* *a* **tellement** *travaillé* **qu'***il* *peut* *se reposer* .
Jean has so much worked that he may have a rest .
Jean has so much worked that he may have a rest.

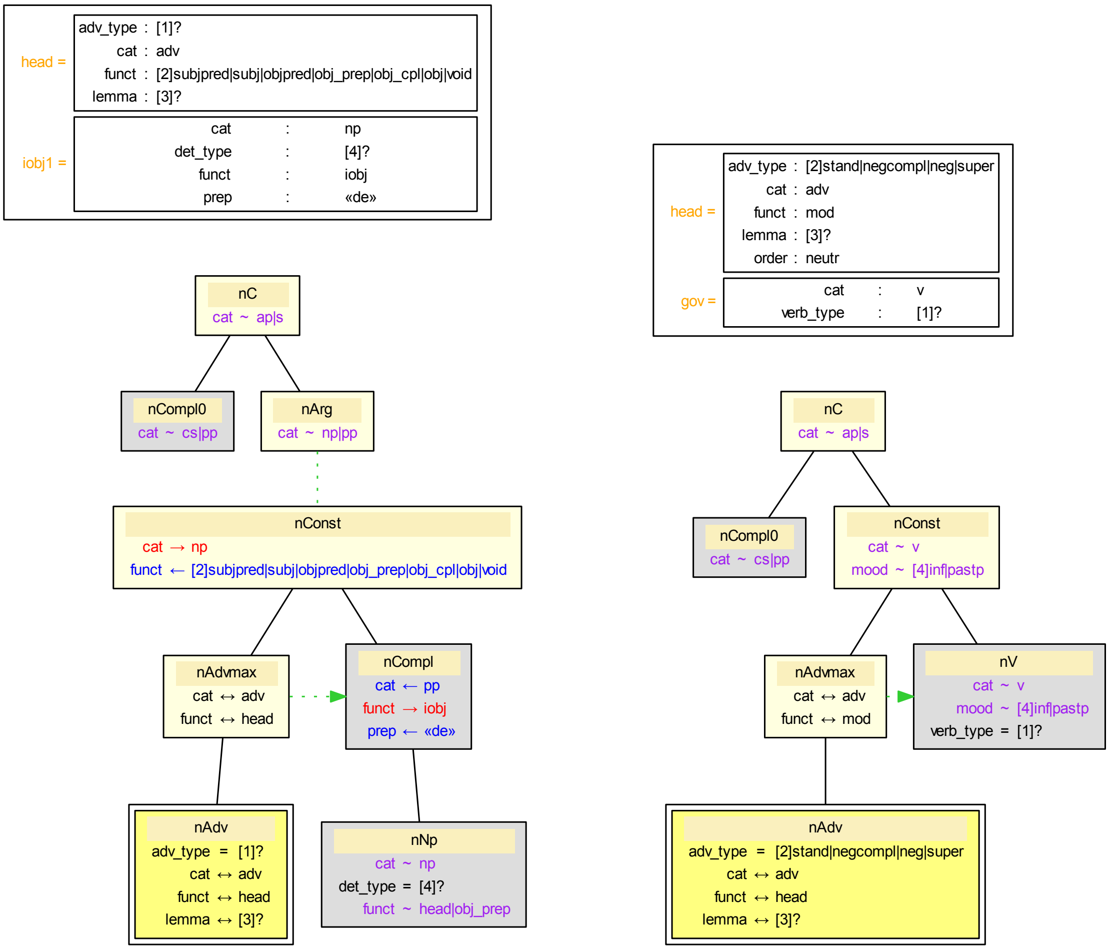

Figure 8.17: PTDs defined by the AdverbWithComplementClause class used in Examples (8.32) and (8.33)

(8.34) *Le* *paysage* *est* **plus** *ensoleillé* *maintenant* **qu'***il* *ne* *l'est* *en*
The landscape is more sunny now than it is in
hiver .
winter .

The landscape is more sunny now than it is in winter.

(8.35) *Le* *paysage* *est* **plus** *ensoleillé* *maintenant* **qu'***en* *hiver* .
The landscape is more sunny now than in winter .
The landscape is more sunny now than in winter.

The AdverbWithComplementClause class defines the common skeleton for all EPTDs. It generates 12 PTDs corresponding to all possible functions of adverbs. Figure 8.17 shows the PTDs corresponding to Examples (8.32) and (8.33).

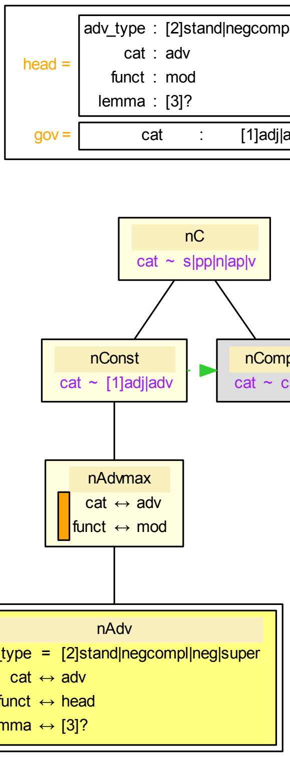

Figure 8.18: PTD defined by the AdverbWithComplementClause class used in Examples (8.34) and (8.35)

In the left PTD, node *nConst* represents the noun phrase made up of the adverb with its complement, *trop d'élèves* in our example. Node *nC* represents the sentence or the adjectival phrase that is the scope of the construction; in our example, it is the whole sentence. The clause that is correlated with the adverb, *pour ne pas venir* in our example, is represented by node *nCompl0* which is a daughter of *nC*. There is an underspecified dominance relation from *nC* to *nConst* because the adverb may be embedded more or less deeply in the clause at which the complement clause is attached. In the right EPTD of Figure 8.17 illustrating Sentence (8.33), there is no such underspecified dominance relation because the adverb modifies a verb which is the head of the clause constituting the scope of the construction.

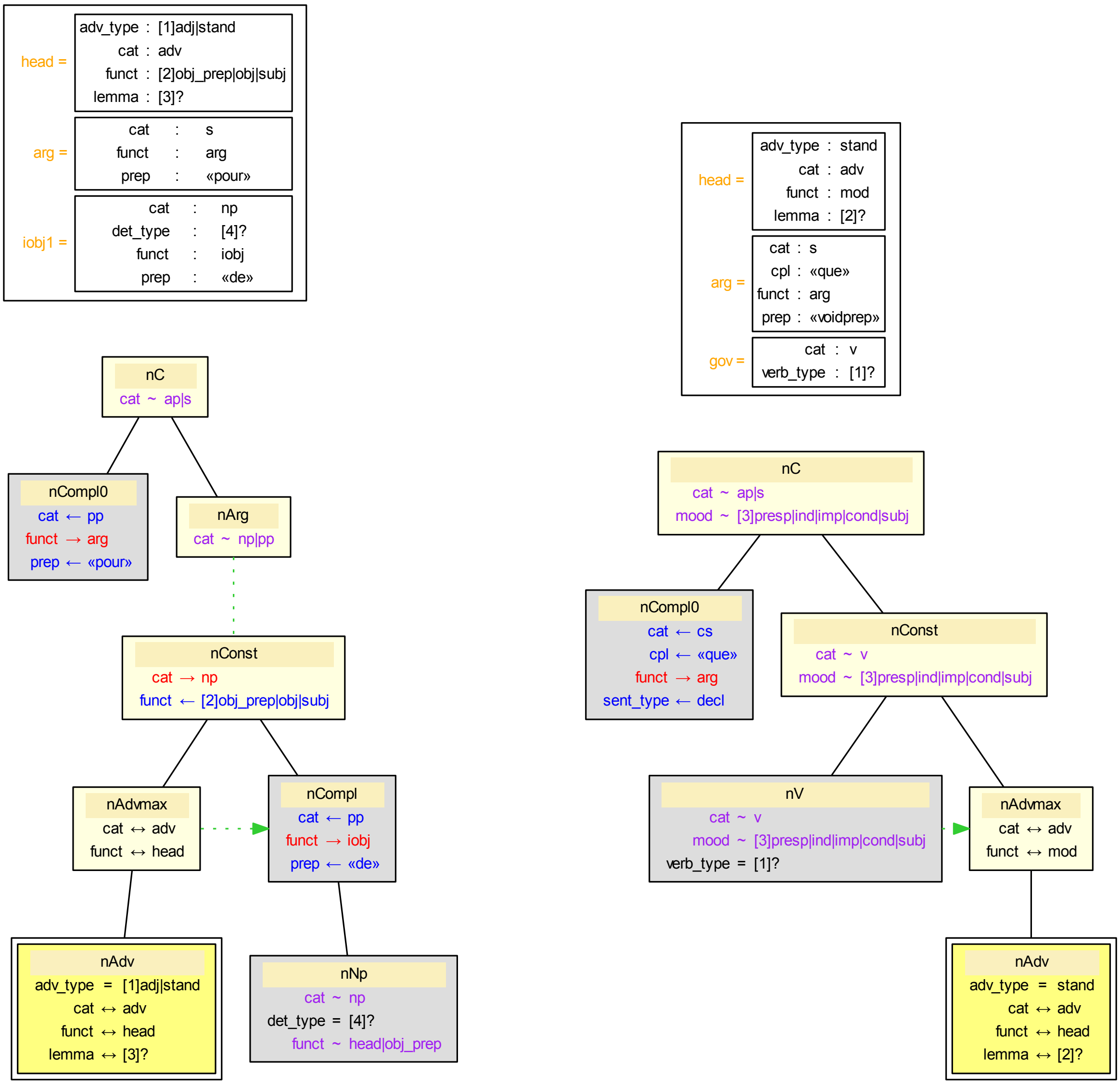

Figure 8.19: EPTDs defined by the ADV_C1_pourS2 and ADV_C1_queS2 classes

It is the same for the EPTD of Figure 8.18 illustrating Sentences (8.34) and (8.35). The scope of the construction is the adjectival phrase that is modified by the adverb.

The AdverbWithComplementClause class is specialised in two sub-classes according to the type of the complement clause that is correlated with the adverb: if it is infinitive, ADV_C1_pourS2 and if it is a finite clause, ADV_C1_queS2.

The ADV_C1_pourS2 class is illustrated with Example (8.32). It defines 10 EPTDs and Figure 8.19 on its left part shows the EPTD used in the example. Node *nCompl0* carries the polarised features `cat` ← `pp`, `prep` ← `pour` and `funct` → `arg` to express that a clause introduced with the preposition *pour* is expected to receive the syntactic function `arg`. In the example, it will be satisfied by *pour ne pas venir*.

The ADV_C1_queS2 class defines 12 EPTDs and Figure 8.19 on its right part shows the EPTD used in Examples (8.34) and (8.35). Node *nCompl* represents the correlated clause. In Sentence (8.34) the clause is complete: it is *qu'il ne l'est en hiver* but in Sentence (8.35), it includes an ellipsis; it reduces to the prepositional phrase *en hiver*.

Chapter 9

Subordinating Words

Surbordinating words are grammatical words used to transform constituents into arguments or adjuncts of predicative expressions. They are divided into prepositions and complementizers and they give rise to two modules of FRIGRAM: PREPOSITION and COMPLEMENTIZER.

9.1 Prepositions

A preposition introduces a noun phrase, an adjectival phrase or a clause to build a prepositional phrase with it.

9.1.1 Interfaces with the lexicon

Prepositions are characterised in interfaces with the feature `head.cat = prep`. Their morphological features and some syntactic properties are gathered in the `head` feature:

- `funct`: it gives the possible functions of the phrases headed by the preposition, which are `arg` (argument in a consecutive construction like *trop beau pour être vrai (too beautiful to be truth)*, `agt` (agent complement in a passive diathesis), `obj_cpl`(object of a complementizer in clauses with ellipsis), `iobj` (indirect object), `mod` (modifier), `obj` (direct object), `objpred` (object predicate), `subjpred` (subject predicate) and `void` (no syntactic function);
- `prep`: this feature is needed by predicate expressions requiring a locative complement without constraints on the locative preposition; in this case, the `prep` feature takes the value `loc`; in other cases, the value of the `prep` feature is the phonological form of the preposition.

A preposition introduces a phrase, the characteristics of which are given by the `obj_prep` feature and described with the following sub-features:

- `cat`: the category of the phrase introduced by the preposition, which can be `n` (common noun), `np` (noun phrase), `ap` (adjectival phrase), `s` (infinitive sentence) or `cs` (complemented sentence);

- `mood`: the mood of the phrase introduced by the preposition, when it is a sentence;
- `cpl`: the complementizer when the phrase introduced by the preposition is a complemented sentence;

9.1.2 The relation between a preposition and its dependent

The skeleton of EPTDs attached at prepositions is defined by the Preposition0 class, which generates the PTD shown in Figure 9.1.

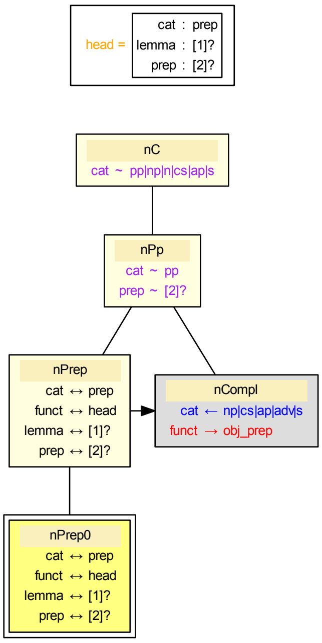

Figure 9.1: The PTD defined by the Preposition0 class

Node *nPrep* represents the preposition with its possible modifiers and node *nPp* represents the prepositional phrase headed by the preposition. This phrase results from the combination of the preposition and the constituent *nCompl* that it introduces. The polarised features `cat` ← `np|cs|ap|adv|s` and `funct` → `obj_prep` means that the preposition expects this constituent to provide it with the function `obj_prep`. Node *nC* represents the constituent that has *nPp* as immediate sub-constituent.

Then, the Preposition0 class is specialised according to the type of its complement *nCompl*. Here are examples illustrating different types of complements.

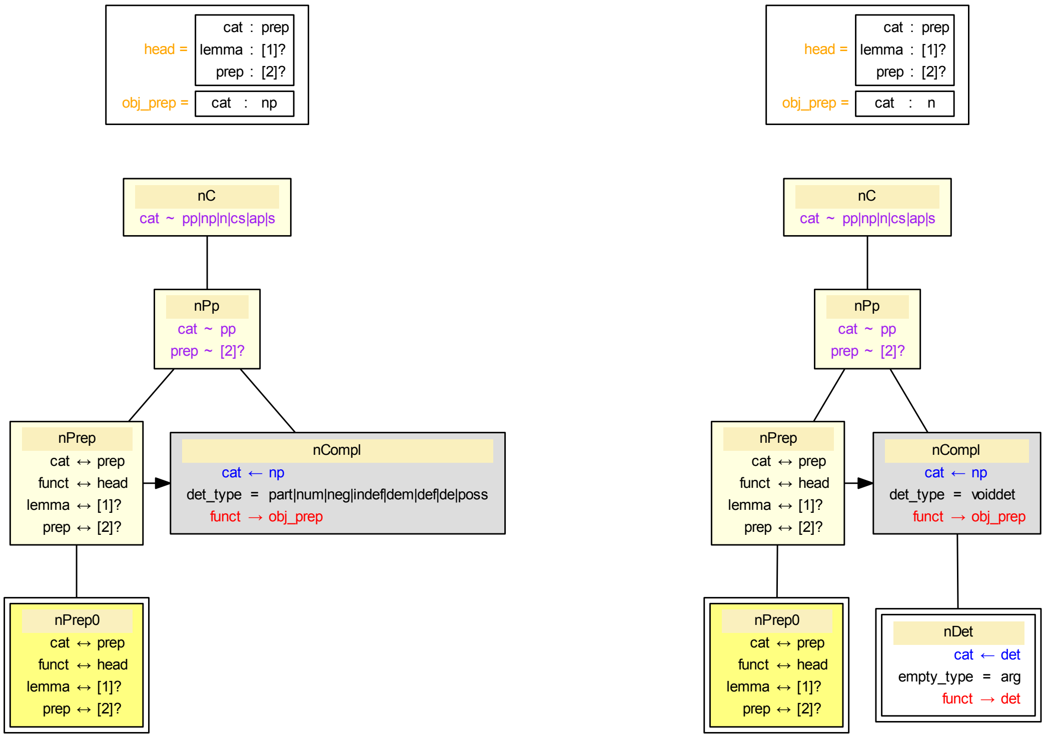
head = cat : prep lemma : [1]? prep : [2]?
obj_prep = cat : np
nC cat ~ pp|np|n|cs|ap|s
nPp cat ~ pp prep ~ [2]?
nPrep cat ↔ prep funct ↔ head lemma ↔ [1]? prep ↔ [2]?
nCompl cat ← np det_type = part|num|neg|indef|dem|def|de|poss funct → obj_prep
nPrep0 cat ↔ prep funct ↔ head lemma ↔ [1]? prep ↔ [2]?
head = cat : prep lemma : [1]? prep : [2]?
obj_prep = cat : n
nC cat ~ pp|np|n|cs|ap|s
nPp cat ~ pp prep ~ [2]?
nPrep cat ↔ prep funct ↔ head lemma ↔ [1]? prep ↔ [2]?
nCompl cat ← np det_type = voiddet funct → obj_prep
nPrep0 cat ↔ prep funct ↔ head lemma ↔ [1]? prep ↔ [2]?
nDet cat ← det empty_type = arg funct → det

Figure 9.2: The PTDs defined by the NounPhrasePreposition and CommonNounPreposition classes

(9.1) *C' est un travail difficile* ***pour*** *Jean .*
It is a job difficult for Jean .
It is a difficult job for Jean.

(9.2) *Jean semble* ***de*** *bonne humeur .*
Jean seems in good mood .
Jean seems to be in a good mood.

(9.3) *C'est* ***à*** *Marie que Jean pense .*
It is of Marie that Jean thinks .
It is of Marie that Jean thinks.

(9.4) *Il passe* ***pour*** *intelligent .*
He looks like clever .
He looks clever.

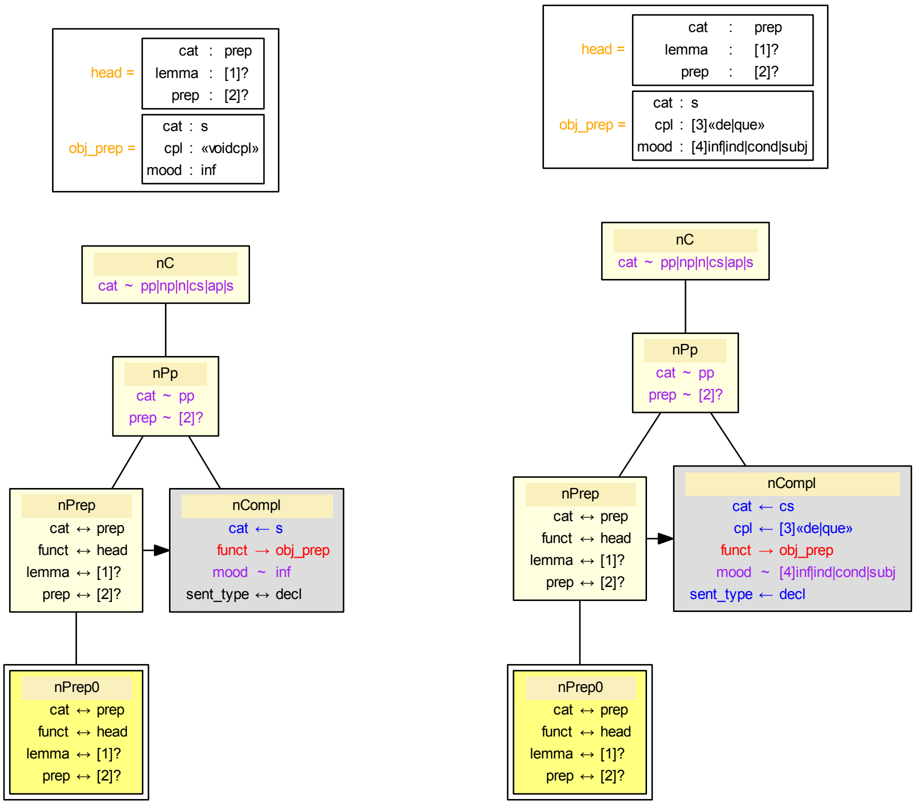

Figure 9.3: The PTDs defined by the CLAUSEPREPOSITION class

(9.5) *Je cherche quelque chose* ***de*** *beau* .
I am looking for something of beautiful .
I am looking for something beautiful.

(9.6) *Jean rêve* ***de*** *venir demain* .
Jean dreams to come tomorrow .
Jean dreams to come tomorrow.

(9.7) *Marie travaille* ***pour*** *que Jean puisse venir* .
Marie works to that Jean can come .
Marie works in order for Jean to be able to come.

(9.8) *Marie travaille* ***en*** *chantant* .
Marie works singing .
Marie works singing.

If this complement is a noun phrase (Sentences (9.1) and (9.3)), the corresponding class is NounPhrasePreposition. If the complement is a common noun (Sentence (9.2)), the corresponding class is CommonNounPreposition. Figure 9.2 shows the PTDs defined by the two classes. For the PTD on the right, an empty node *nDet* represents the missing determiner.

When the complement is an adjectival phrase (Sentences (9.4), (9.5) and (9.8)), the Preposition0 class is specialised in the AdjectivalPhrasePreposition class. This class differs from the NounPhrasePreposition class only on features attached at node *nCompl*: `cat` ← `np` is replaced with `cat` ← `ap`, `det_type` is not present and there is an feature `mood` ∼ `pastp|presp|voidmood`.

When the complement is a clause, the Preposition0 class is specialised in the ClausePreposition class, which defines two PTDs, according to the type of the clause: simple infinitive clause (Sentence (9.6)) or complemented clause (Sentence (9.7)). Figure9.3 shows both PTDs. There is a difference between the two PTDs on the polarity of feature `sent_type` for node *nCompl*: since an infinitive is neutral with respect to the type of the clause, the feature is saturated, whereas, for a complemented clause, the type of the clause is determined by the complementizer; therefore in this case, the polarity of `sent_type` is negative.

The four classes NounPhrasePreposition, CommonNounPreposition, AdjectivalPhrasePreposition and ClausePreposition are gathered by disjunction in a unique class Preposition.

9.1.3 The different functions of the prepositional phrase

Then, the Preposition class is specialised according to the syntactic function of the prepositional phrase *nPp* in the constituent *nC*. If it is a complement required by the head of *nC* (Sentence (9.3)), the specialised class is Preparg_C1.

The class adds polarities to features of Node *nCompl*, which become `cat` → `pp`, `prep` → `?` and `funct` ← `subjpred|objpred|obj_cpl|iobj|arg|agt|void`.

There is a similar case illustrated with Sentences (9.2) and (9.3). The prepositional phrase is a complement required by a verb but it plays the role of a direct predicate complement: the preposition is not constrained by the verb. This occurs in two circumstances: when the preposition allows the prepositional phrase to play the role of a predicate complement as in Example (9.2), and when the prepositional phrase is an extracted constituent in a cleft sentence, as in Example (9.3).

This case is taken into account by the class PREPpred_C1, which defines 10 EPTDs. Figure 9.4 shows two EPTDs used in the parsing of Sentences (9.2) and (9.3).

The AdjunctPreposition class defines PTDs for prepositions introducing adjunct complements. It inherits the Preposition class and adds saturated features `cat` ↔ `pp`, `funct` ↔ `mod` and `prep` ↔ `?` to node *nPp*. The EPTD used in this case is shown in Figure 9.5.

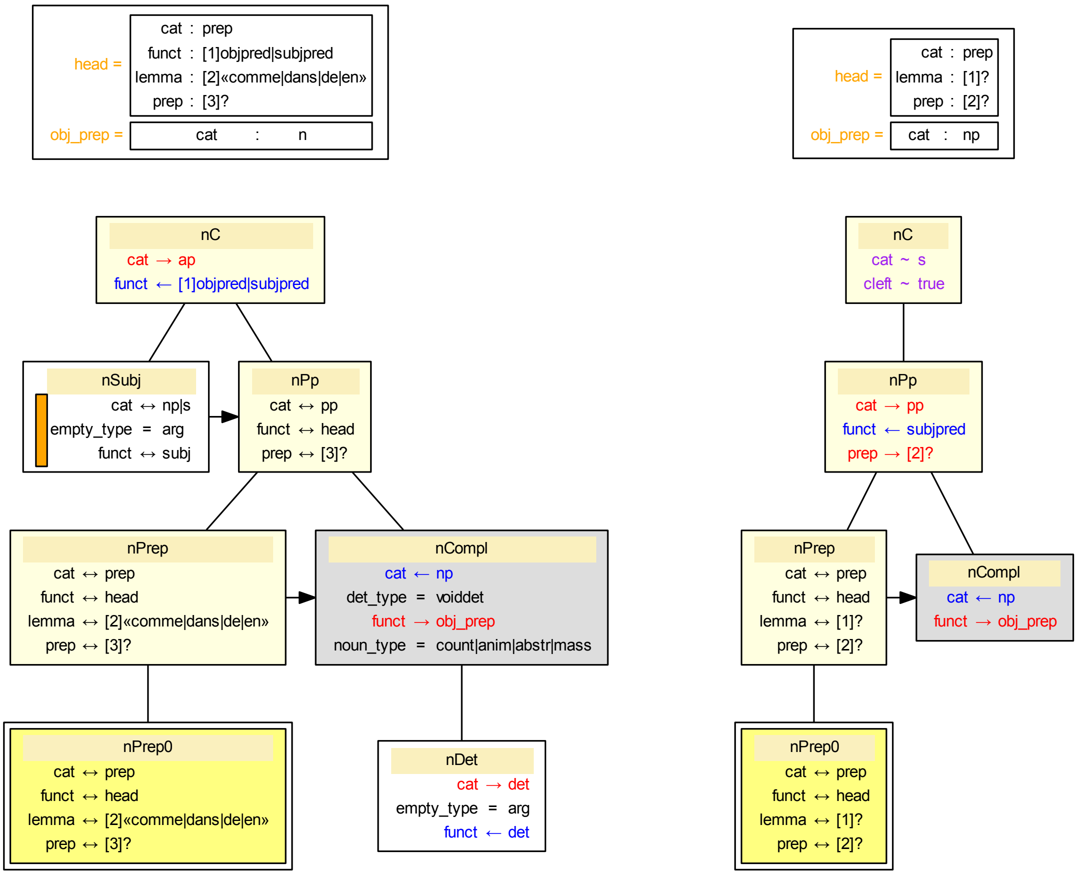

Figure 9.4: EPTDs defined by the PREPPred_C1 class

The AdjunctPreposition class is specialised in three subclasses, according to the category that is modified by the prepositional phrase: PREPAdjn_C1 for nouns (Example (9.5)), PREPAdjap_C1 for adjectival phrases (Example (9.1)), PREPAdjs_C1 for sentential phrases (Examples (9.7) and (9.8)). The last class distinguishes three cases according to the category of the phrase introduced by the preposition: noun phrase, clause or gerundive. For gerundives, the EPTD expresses that the subject of the participle introduced by the preposition is the subject of the clause modified by the prepositional phrase. For instance, in Sentence (9.8), the subject of *entrant* is the subject of the main verb *travaille*.

The two examples below illustrate a particular use of the preposition *à* followed by a transitive infinitive. The prepositional phrase they build is used as an adjective and the object of the transitive infinitive is the subject of the adjective.

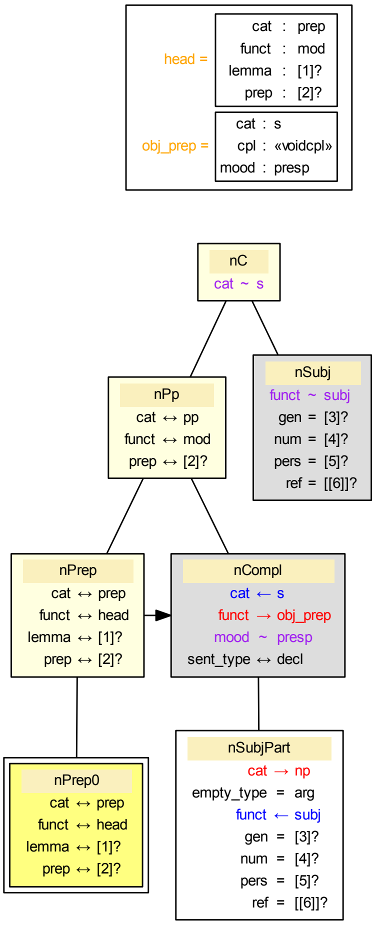

Figure 9.5: EPTD defined by the PREPADJS_C1 class and used in Sentence (9.8)

The PREPA_S1INF class defines the EPTDs associated with the preposition *à* in this case and shown in Figure 9.6. In both EPTDs, Nodes *nSubj* and *nObj* co-refer to the same entity.

9.2 Complementizers

Complementizers are linking words used to transform clauses into arguments of predicate expressions. The words *à* and *de* are usually considered as prepositions but here they are considered as complementizers when they introduce infinitives that are direct objects or predicate complements. Here are examples illustrating different uses of complementizers (marked in bold).

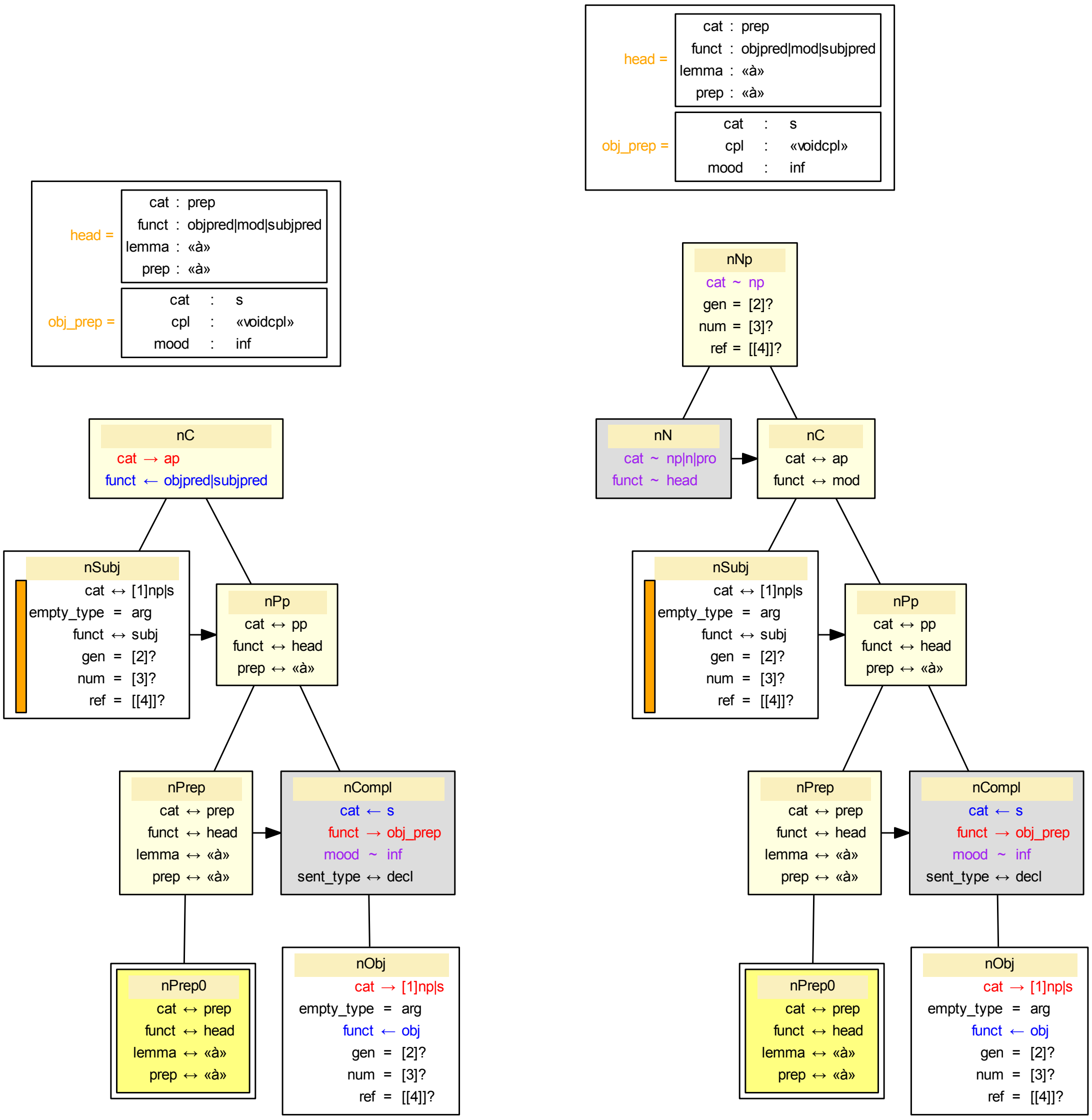

Figure 9.6: EPTD defined by the PrepA_S1Inf class and used in Sentences (9.9) and (9.10)

(9.9) *La maison est **à** vendre .*
The house is to sell .
The house is to sell.

(9.10) *C' est un livre **à** lire .*
That is a book to read .

That is a book to read.

(9.11) *Jean croit* ***que*** *Marie viendra .*
Jean believes that Marie will come .
Jean believes that Marie will come.

(9.12) *L'idée* ***que*** *Marie vienne me réjouit .*
The idea that Marie comes delights me .
The idea that Marie comes delights me.

(9.13) *C' est dans l' après-midi* ***que*** *Jean arrivera .*
It is in the afternoon that Jean will arrive .
It is in the afternoon that Jean will arrive.

(9.14) *Jean décide* ***de*** *partir .*
Jean decides to go .
Jean decides to go.

(9.15) ***Que*** *Jean parte !*
that Jean goes !
that Jean goes!

(9.16) *Jean demande* ***si*** *Marie part .*
Jean asks if Marie goes .
Jean asks if Marie goes.

(9.17) *Marie part* ***si*** *Jean vient .*
Marie will go if Jean is coming .
Marie will go if Jean is coming.

(9.18) *Jean marche* ***comme*** *quand il a bu .*
Jean is walking like when he has drunk .
Jean is walking like when he has drunk.

(9.19) *Jean est un homme* ***comme*** *il en existe peu .*
Jean is a man as it exists few .
Jean is a man as a few exists.

(9.20) *Jean marche moins vite* ***que*** *pour aller au travail .*
Jean is walking less fast than to go to work .
Jean is walking less fast than to go to work.

9.2.1 Interfaces with the lexicon

Complementizers are characterised in interfaces with the feature `head.cat = cpl`. Their morphological features and some syntactic properties are gathered in the `head` feature:

- `cpl`: this feature has the phonological form of the complementizer as value, which is also represented with feature `lemma` and it seems to be redundant, but in the EPTD anchored by the complementizer, the feature is shared by the node represented the complemented clause. Because of limitations in the current version of FRIGRAM, a feature cannot share its value with another type of feature. Feature `cpl` of the complemented clause cannot share its value with feature `lemma` of the complementizer. In the interface, we need to repeat the value in a feature `cpl`.
- `funct`: the syntactic function of the complemented clause can take the values `app` (apposition), `arg` (in a comparative construction, depending on the adjective or the adverb introducing the comparison as the second term of the comparison - see Sentence (9.20)), `dis` (dislocation), `iobj` (indirect object), `mod` (modifier), `mod_cleft` (introducing the subordinated clause in a cleft construction), `obj` (direct object), `obj_prep` (object of a preposition), `subj` (subject), `subjpred` (predicate complement referring to the subject), `void` (no syntactic function);

A complementizer introduces a clause, the characteristics of which are given by the `clause` feature and described with the following sub-features:

- `mood`: the mood of the clause introduced by the complementizer;
- `sent_type`: the type of the clause can take the values `decl` (declarative) or `inter` (interrogative).

When a complemented clause is a modifier, the `gov` feature indicates the category of the constituent that is modified, with the sub-feature `cat`, which can take the values `s` or `np`.

9.2.2 The different functions of complementizers

The Complementizer class provides the common skeleton of the EPTDs for all complementizers. It defines the PTD shown in Figure 9.7. In this PTD, node *nS* represents the clause without its complementizer and *nCs* the clause with its complementizer.

Standard use of complementizers

Then, the Complementizer class is specialised in different subclasses. Three of them correspond to ordinary functions of complemented clauses.

- CPLarg_S1 defines the EPTD associated with complementizers introducing clauses that are arguments of predicative expressions (Examples (9.11), (9.14) and (9.16)) or that give an imperative or exclamative feature to full sentences (Example (9.15));

- CPLADJ_S1 defines the EPTD associated with complementizers introducing clauses that are adjuncts of sentences (Example (9.17)), noun phrases (Example (9.19)) or adjectival phrases;

- CPLQUE-ADJN_S1 defines the EPTD associated with the complementizer *que* introducing clauses that are adjuncts of common nouns (Example (9.12));

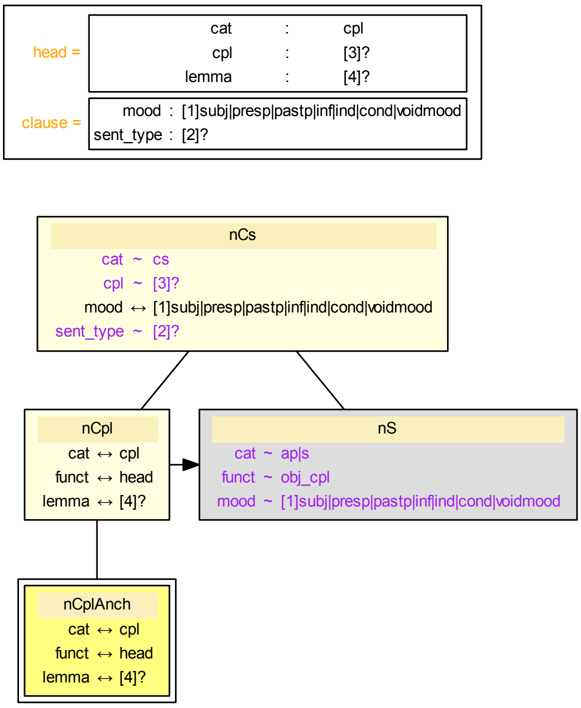

Figure 9.7: The PTD defined by the COMPLEMENTIZER class

The CPLARG_S1 class defines the EPTD shown in Figure 9.8. Node *nS* has a saturated feature `sent_type` ↔ `decl`, whereas Node *nCs* carries a positive feature `sent_type`, the value of which depends on the nature of the complementizer. A positive polarity is added to the feature `cpl` with a restriction on its possible values. The `cat` and `funct` features of Nodes *nS* and *nCs* are polarised as expected.

Figure 9.9 shows the EPTDs defined by the classes CPLADJ_S1 and CPLQUE-ADJN_S1. In both EPTD, a saturated feature `funct` ↔ `mod` attached at node *nCs* expresses that the associated complemented clause is a modifier of the head of the mother constituent. In the right EPTD, a node *nN* expresses the constraint that the head of *nNp* is a common noun. In this way, the following sentence is rejected as ungrammatical because *Jean* is a proper noun.

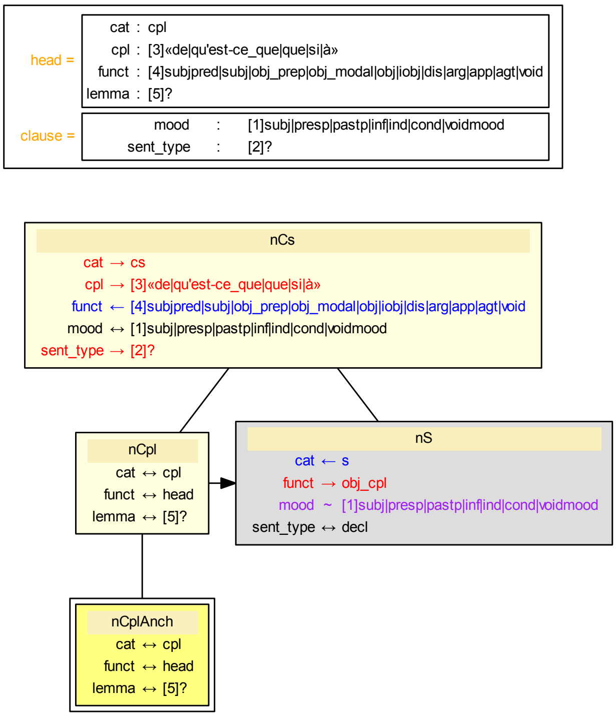
head =
cat : cpl
cpl : [3]«de|qu'est-ce_que|que|si|à»
funct : [4]subjpred|subj|obj_prep|obj_modal|obj|iobj|dis|arg|app|agt|void
lemma : [5]?
clause =
mood : [1]subj|presp|pastp|inf|ind|cond|voidmood
sent_type : [2]?
nCs
cat → cs
cpl → [3]«de|qu'est-ce_que|que|si|à»
funct ← [4]subjpred|subj|obj_prep|obj_modal|obj|iobj|dis|arg|app|agt|void
mood ↔ [1]subj|presp|pastp|inf|ind|cond|voidmood
sent_type → [2]?
nCpl
cat ↔ cpl
funct ↔ head
lemma ↔ [5]?
nS
cat ← s
funct → obj_cpl
mood ~ [1]subj|presp|pastp|inf|ind|cond|voidmood
sent_type ↔ decl
nCplAnch
cat ↔ cpl
funct ↔ head
lemma ↔ [5]?

Figure 9.8: The EPTD defined by the CPLArg_S1 class

(9.21) **Jean* ***que*** *c' est un homme bien vient aujourd'hui .*
Jean that it is a man good is coming today .
*Jean that is a good man is coming today.

CPL_S1cleft defines the EPTD associated with complementizers introducing subordinated clauses related to cleft construction (Example (9.13)); their study is postponed to Chapter 10.

Complementizers in comparisons

The three first following examples are a resumption of previous examples.

(9.22) *Jean est un homme* ***comme*** *il en existe peu .*
Jean is a man as it exists few .
Jean is a man as a few exists.

(9.23) *Jean* *marche* ***comme*** *[il* *marche]* *quand* *il* *a* *bu* *.*
Jean is walking like [he walks] when he has drunk .
Jean is walking like [he walks] when he has drunk.

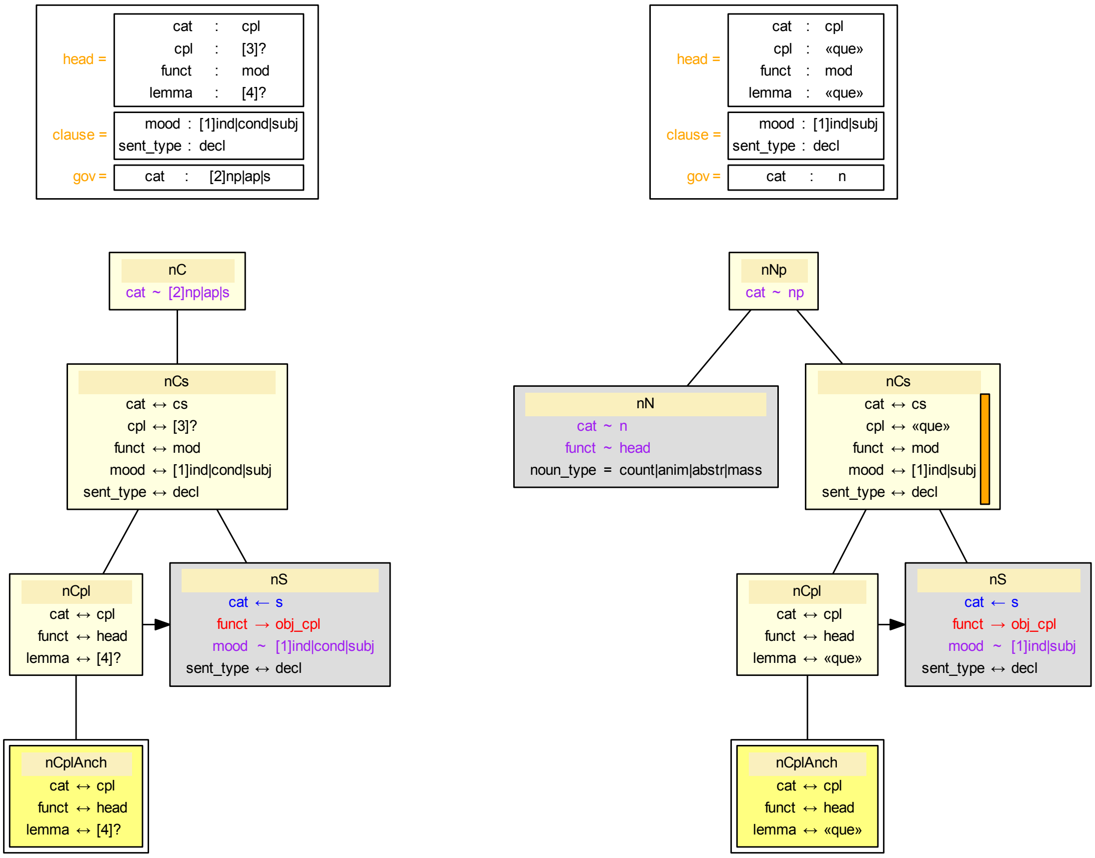

Figure 9.9: The EPTDs defined by the CPLadj_S1 and CPLque-adjn_S1 classes

(9.24) *Jean* *marche* *moins* *vite* ***que*** *[il* *marche]* *pour* *aller* *au* *travail* *.*
Jean is walking less fast than [he walks] to go to work .
Jean is walking less fast than [he walks] to go to work.

(9.25) *Jean* *fait* **comme** *il* *le* *pense* *.*
Jean does as he it thinks .
Jean does as he thinks it.

(9.26) *Jean* *est* **comme** *Marie* *[est]* .
Jean is like Marie [is] .
Jean is like Marie.

The complementizers *comme* and *que* are used in comparisons. When *que* introduces complete clauses (Example (9.22)), it is considered as a standard complementizer, and the corresponding EPTD is defined by the standard class CPLARG_S1. Nevertheless, *que* often introduces a clause with ellipsis, as Example (9.24) show it (The elided expression appears between square brackets). A specific class, inheriting the COMPLEMENTIZER class, takes this case into account: the CPLCOMPAR_C1 class. This class defines two EPTDs shown in Figure 9.10. The difference between them lies in the role of the constituent *nC* with respect to the elided verb *nVmax*: required argument in the left EPTD or modifier in the right EPTD.

Class CPLCOMME_S1, inheriting the COMPLEMENTIZER class, defines the 6 EPTDs anchored by *comme* according to two dimensions:

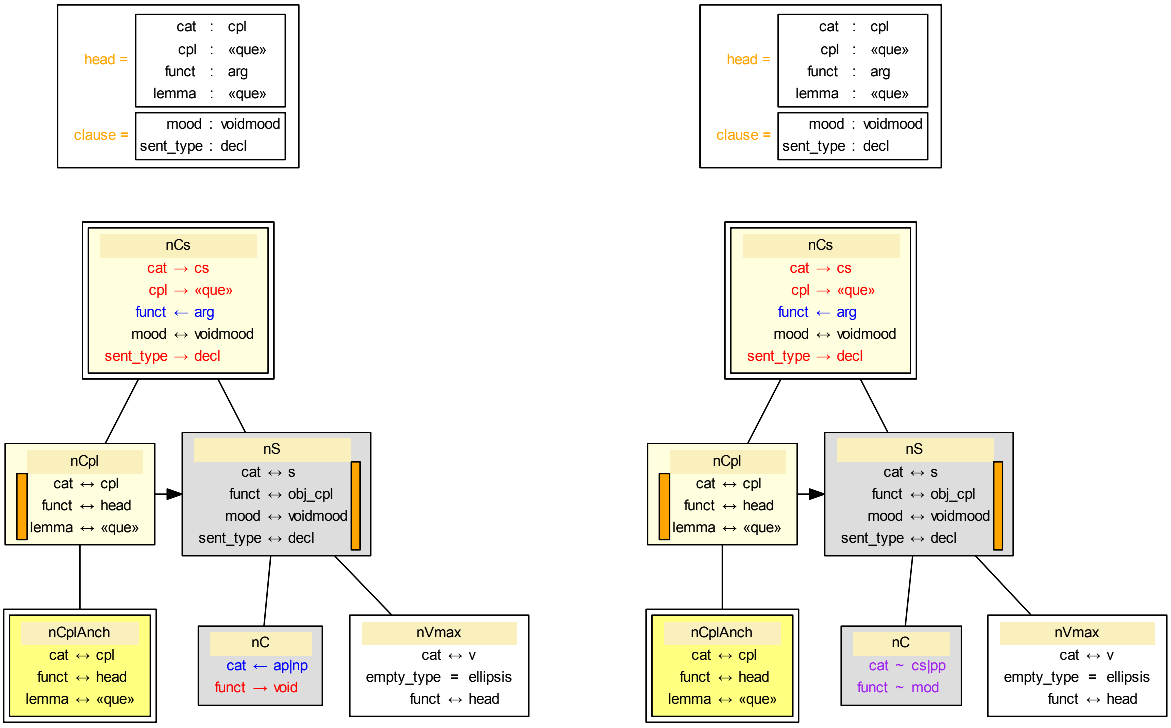

Figure 9.10: The EPTDs defined by the CPLCOMPAR_C1 class

- the clause introduced with *comme* is the required complement of a verb (Examples (9.25) and (9.26)) or it is an adjunct of another constituent (Examples (9.22) and (9.23)).

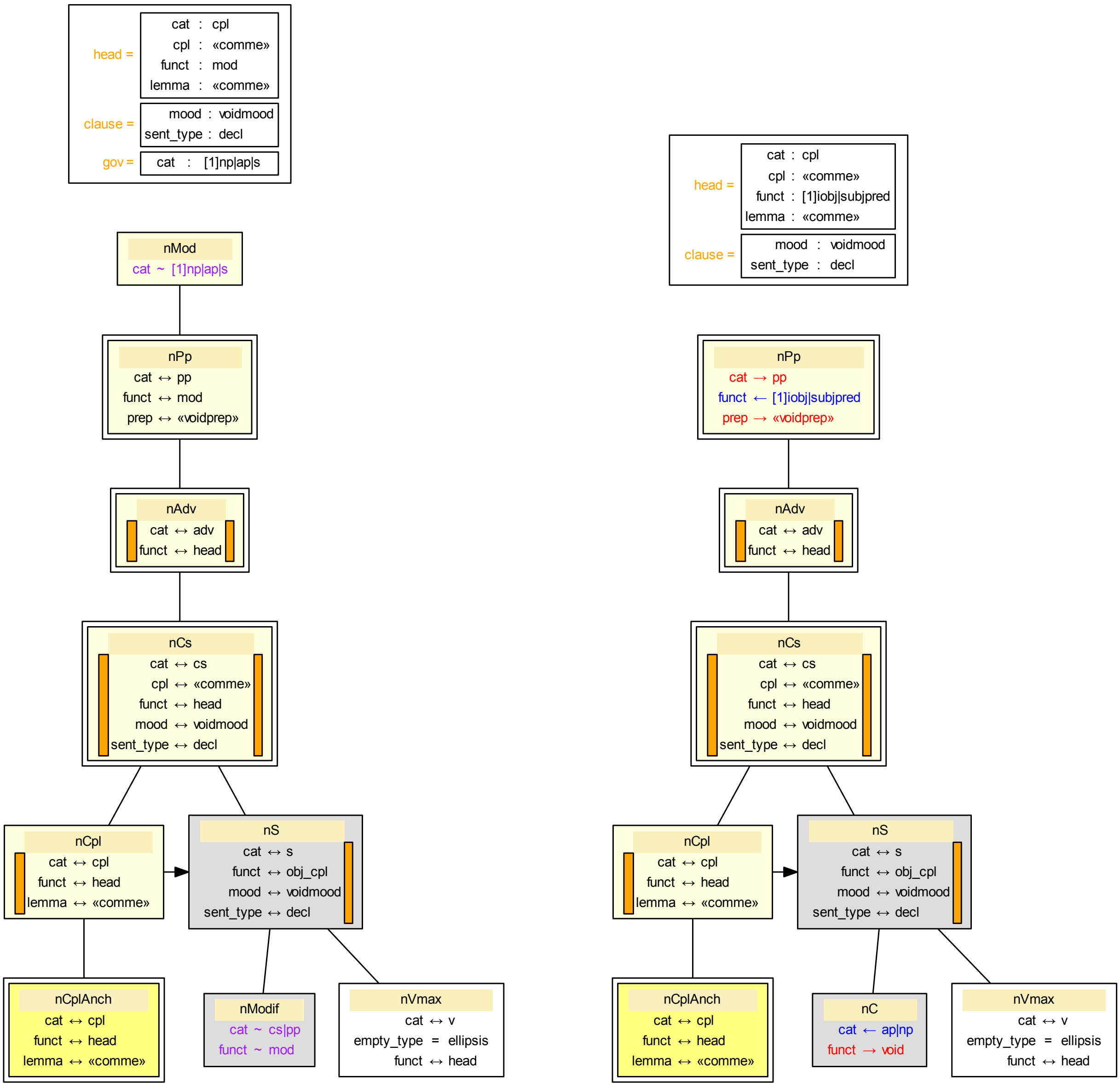

Figure 9.11: EPTDs defined by the CPLcomme_S1 class

- the clause introduced with *comme* is a complete clause (Examples (9.22) and

(9.25)), or it contains an ellipsis: either it reduces to and adjunct (Example (9.23)), or it reduces to a noun phrase (Example (9.26)) or an adjectival phrase.

Figure 9.11 shows two of the 6 EPTDs, those used in the parsing of Examples (9.23) and (9.26).

For both EPTDs, node *nCs* represents the complemented clause. As the clause behaves as an adverb, this node has a mother node *nAdv*, which is an adverb, and which has a mother node *nPp*, because this adverb plays the same role as a prepositional phrase.

There are two differences between the EPTDs:

- In the left EPTD, node *nPp* represents a modifier, whereas in the right EPTD, node *nPp* represents a required complement, which can be an indirect object or a subject predicative complement.

- In the left EPTD, the clause introduced by *comme* reduces to an adjunct represented with node *nModif*, and in the right EPTD, this clause reduces to a noun phrase or an adjectival phrase.

Chapter 10

Extraction

Relative clauses, cleft clauses and interrogative clauses with partial interrogation give rise to extraction of constituents. These constituents are put at the beginning of the clause from which they are extracted and in our approach, a trace is left at the initial place under the shape of an empty constituent. In the following examples, the extracted constituent is in bold and the trace is marked with the □ symbol.

(10.1) **L'ingénieur** *que connaît Marie* □ *arrive demain .*
the-engineer whom knows Marie □ is-arriving tomorrow .
the engineer whom Marie knows is arriving tomorrow.

(10.2) **Quel lieu** *Marie pense-t-elle que Jean souhaite visiter* □ *demain*
which place Marie believe that Jean hope to visit tomorrow
?
?
which place does Marie believe that Jean hope to visit tomorrow?

(10.3) *C'est* **à Paris** *que Marie souhaite aller* □ *demain .*
It's to Paris that Marie hopes to go tomorrow .
It's to Paris that Marie hopes to go tomorrow.

(10.4) *L'ingénieur* **qui** *connaît Marie arrive demain .*
the-engineer who knows Marie is-arriving tomnorrow .
The engineer who knows Marie is arriving tomorrow.

In Sentence (10.1), the relative clause *que connaît Marie* is a simple clause. At the opposite, in Sentence (10.2), the interrogative clause *Quel lieu Marie pense-t-elle que Jean souhaite visiter demain* is a complex clause with an embedded object clause *que Jean souhaite visiter demain.* This clause is also a complex clause with an embedded infinitive *visiter demain*, which is the source of the extracted object *quel lieu.* Example (10.3) shows an example of cleft clause. In presence of a subject relative or interrogative pronoun, as in Example (10.4), there is no extraction because the pronoun is in situ: it occupies the canonical position of the constituent that it replaces. In our example, the relative pronoun *qui* occupies the position of the subject of *connaît*, before it.

10.1 Module ExtractGramWord

The common ability of some grammatical words to express extraction is represented with the ExtractGramWord module. The basic class of the module, ExtractedComplement0, defines the PTD shown in Figure 10.1. It introduces three clause levels:

- *nS* represents the complete relative, interrogative or object clause that represents the scope of the concerned grammatical word; the subject of this clause is *nSubj* and the head verb is *nVmax*;
- *nS0* represents the clause that immediately includes the trace of the extracted word; its subject is *nSubj0*;
- *nS1* represents an intermediate clause between *nS* and *nS0*; when *nS0* identifies with *nS*, it also identifies with *nS1*; otherwise, it represents the most external clause that is embedded in *nS*.

Node *nExtract* represents the extracted constituent and *nTrace*, which is an empty node, its trace.

In the interface, feature `clause` gives some properties of the clause *nS0*, which is the source of the extraction. So, feature `extract.mood` is co-indexed with feature `mood` of node *nS0*. Still in the interface, feature `extract` gives some properties of the extracted constituent. That is why feature `extract.cat` is co-indexed with feature `cat` of node *nTrace*.

For Sentence (10.1), in the PTD associated with *que*, nodes *nS*, *nS1* and *nS0* are merged and represent the relative clause *que connaît Marie*. Node *nTrace* represents the trace of the object of *connaît*.

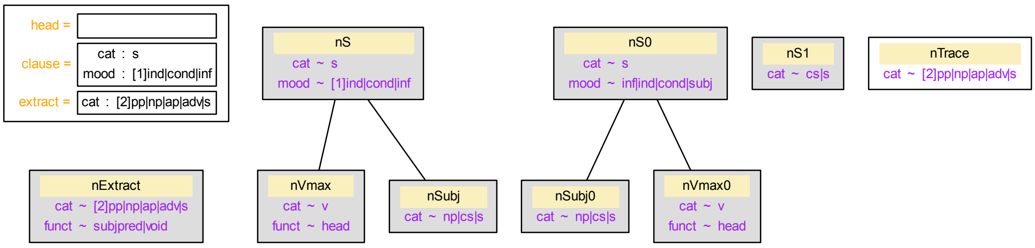

Figure 10.1: The PTD defined by the ExtractedComplement0 class

For Sentence (10.2), in the PTD associated with *quel*, node *nS* represents the interrogative clause *Quel ingénieur Marie pense-t-elle que Jean souhaite rencontrer demain* and node *nS0* represents the object infinitive *rencontrer demain.* Node *nS1* represents the intermediate clause *que Jean souhaite rencontrer demain.* Node *nTrace* represents the trace of the object of *rencontrer.*

For Sentence (10.3), in the PTD associated with *que*, node *nS* represents the subordinated clause *que Marie souhaite aller demain.* Nodes *nS1* and *nS0* identify and represents the clause *aller demain.* Node *nTrace* represents the trace of the indirect object of *aller.*

10.1.1 Verb subject order in the clause that is the location of the trace

In *nS0*, the relative order between the verb and the subject depends on two parameters: the nature of the grammatical word and the transitive feature or not of the verb[1]. Here are examples that illustrate the influence of the two parameters.

(10.5) *L'ingénieur que connaît Marie* □ *arrive demain .*
the-engineer whom knows Marie □ is-arriving tomorrow .
The engineer whom Marie knows is arriving tomorrow.

(10.6) *L'ingénieur que Marie connaît* □ *arrive demain .*
the-engineer whom Marie knows □ is-arriving tomorrow .
The engineer whom Marie knows is arriving tomorrow.

(10.7) *Que connaît Marie* □ *?*
what know Marie □ ?
What does Marie know?

(10.8) **Que Marie connaît-elle* □ *?*
what Marie know □ ?

(10.9) *Quel est cet homme* □ *?*
which is that man □ ?
Which is that man?

(10.10) *L'entreprise de laquelle Marie sort* □ *va fermer .*
The-firm from which Marie is-going-out □ is-going to-be-closed .
The firm from which Marie is going out is going to be closed.

(10.11) *L'entreprise de laquelle sort Marie* □ *va fermer .*
The-firm from which is-going-out Marie □ is-going to-be-closed .
The firm from which Marie is going out is going to be closed.

[1] When *nS* is different from *nS0*, the verb subject order in *nS* is the canonical order.

(10.12) *Comment Marie va-t-elle* □ *?*
how Marie feels □ ?

How does Marie feel?

(10.13) *Comment va Marie* □ *?*
how feel Marie □ ?

How does Marie feel?

(10.14) *Comment Marie a-t-elle rencontré Pierre* □ *?*
how Marie did-she meet Pierre □ ?

How did Marie meet Pierre?

(10.15) **Comment a rencontré Pierre Marie* □ *?*
how did meet Pierre Marie □ ?

how did Marie meet Pierre?

Examples (10.5) and (10.6) show that the order between verb and subject is free for the relative pronoun *que*[2]. Examples (10.7) and (10.8) show that the only inverted subject is allowed for the interrogative pronoun *que*. That is the same for the interrogative pronoun *quel* (Example (10.9)). For the relative pronoun *laquelle* and the interrogative adverb *comment*, the order is free as Examples (10.10), (10.11), (10.12) and (10.13) show it. The presence of a transitive verb forces the subject to precede the verb, as Examples (10.14) and (10.15) show it.

Within our grammar, we cannot represent the influence of transitivity on the order between the verb and the subject, because there is no way of expressing the object absence for the verb in the PTD of the grammatical word.

We have only represented the influence of the nature of the grammatical word on this order and we have realized it via the PTD interfaces. Every interface has a feature `clause.subj`, which expresses the order between the verb and the subject in *nS0*, when it is a finite clause:

- it has the value *can* if the order is the *canonical* order subject-verb;
- it has the value *rev* if the order is the *reverse* order verb-subject.

Another factor makes the management of the subject-verb order more complicated, the combination of the head verb of *nS* with a sequence of infinitives.

(10.16) *Comment pense pouvoir venir Jean* □ *?*
how believe can come Jean □ ?

How does Jean believe to be able to come?

[2] This is not completely true because the inversion is allowed if the distance between the verb and the subject is not too large. For instance, the following sentence is not grammatical: **L'ingénieur que connaît depuis qu'elle l' a rencontré Marie arrive demain.*

(10.17) *Comment Marie pense-t-elle que pourra venir Jean □ ?*
how Marie believes that can come Jean □ ?
How does Marie believe to be able to come?

(10.18) * *Comment pense que pourra venir Jean □ Marie ?*
how believes that can come Jean □ Marie ?

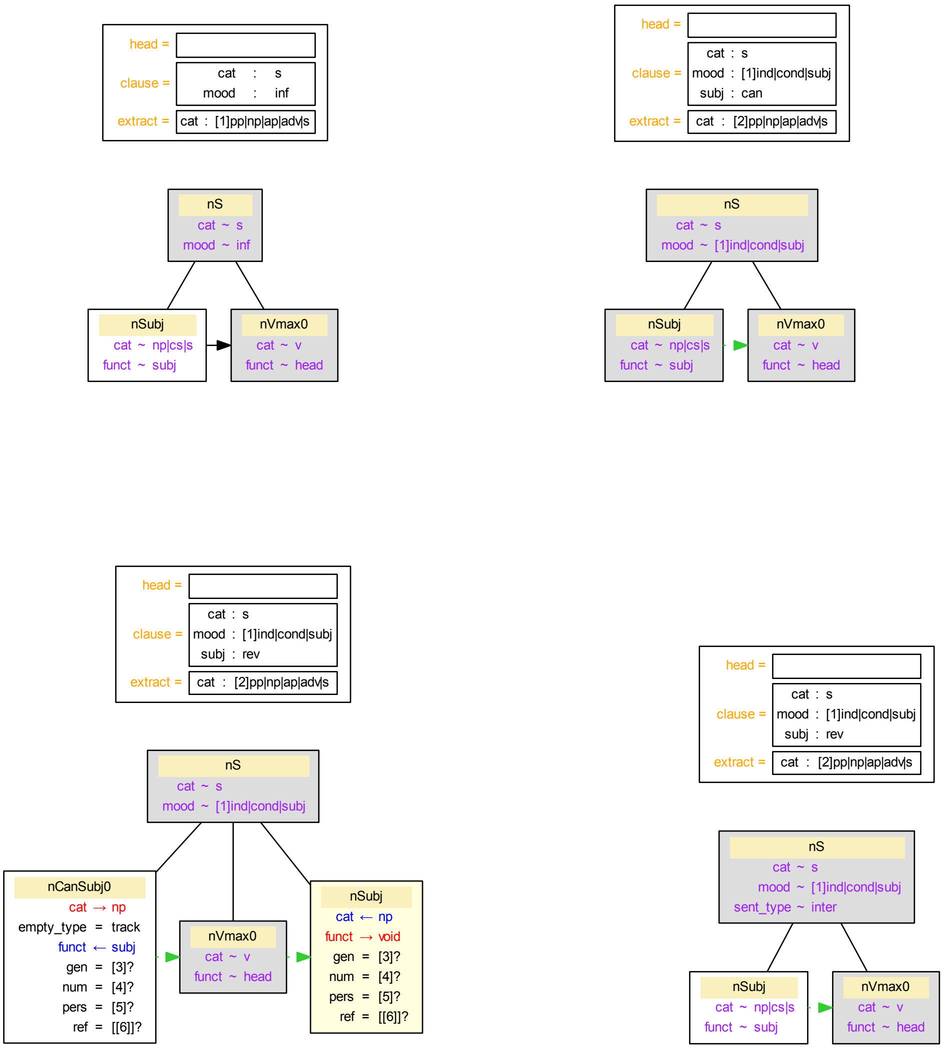

Figure 10.2: The differences between the PTDs defined by the ImmediateExtractedConstituent class

Example (10.16) shows that the sequence *pense pouvoir venir* behaves as a single verb and it allows subject-verb inversion. Examples (10.17) and (10.18) show that such an inversion is not allowed when the object clause is a finite clause, like *que pourra venir Jean*. Because of this complication, we distinguish two cases:

- the clauses *nS*, *nS1* and *nS0* identify; this case is dealt with by the ImmediateExtractedComplement class;
- clause *nS* strictly includes clause *nS0*, that case is dealt with by the DistantExtractedComplement class.

The ImmediateExtractedComplement class inherits the basic class *ExtractedConstituent0*. Then, it distinguishes four cases according to the mood of *nS* and to the constraint imposed by the grammatical word responsible of the extraction. Therefore, it generates four PTDs, the differences of which are shown by Figure 10.2. In the order:

- The first PTD corresponds to a clause *nS* that is an infinitive. We must deal with this case apart because the verb-subject order has no sense for an infinitive and the sentence has only one parse tree, even if the grammatical word responsible of the extraction is neutral with respect to the order verb-subject (for instance *où aller ?*).
- The second PTD corresponds to a finite clause *nS*, where the order verb-subject is canonical (Examples (10.7), (10.11), (10.13) and (10.15)).
- The third PTD corresponds to a finite clause *nS*, where the order verb-subject is inverted and where the subject is full (Examples (10.6), (10.8), (10.8), (10.10), (10.12) and (10.14)). The real subject is represented with the full node *nSubj* and it has no syntactic function, which is expressed with the feature `funct` → `void`. The function `subj` is carried by an empty trace, which is represented with node *nCanSubj0*. This node is in canonical position and it co-refers with *nSubj*.
- The last PTD corresponds to a finite clause *nS*, where the order verb-subject is inverted and where the subject is empty (for instance in the sentence *comment va-t-il ?*). We need to distinguish this case from the previous one because the subject is not represented by a full noun phrase. As it was explained in Section 6.2.2 of Chapter 6, a subject clitic is always associated with an empty trace in the canonical position of a subject, regardless its position with respect to the verb.

The DistantExtractedComplement class inherits the basic class *ExtractedConstituent0*. Now, the extraction is distant and node *nS1* is an immediate sub-constituent of *nS*, whereas there is an underspecified dominance from *nS1* to *nS0*. Then, we distinguish three cases, which are related not only to the mood of *nS* but also to the mood of *nS0*. Contrary to the previous class, they are not related to the constraint imposed by the grammatical word responsible of the extraction on verb-subject order.

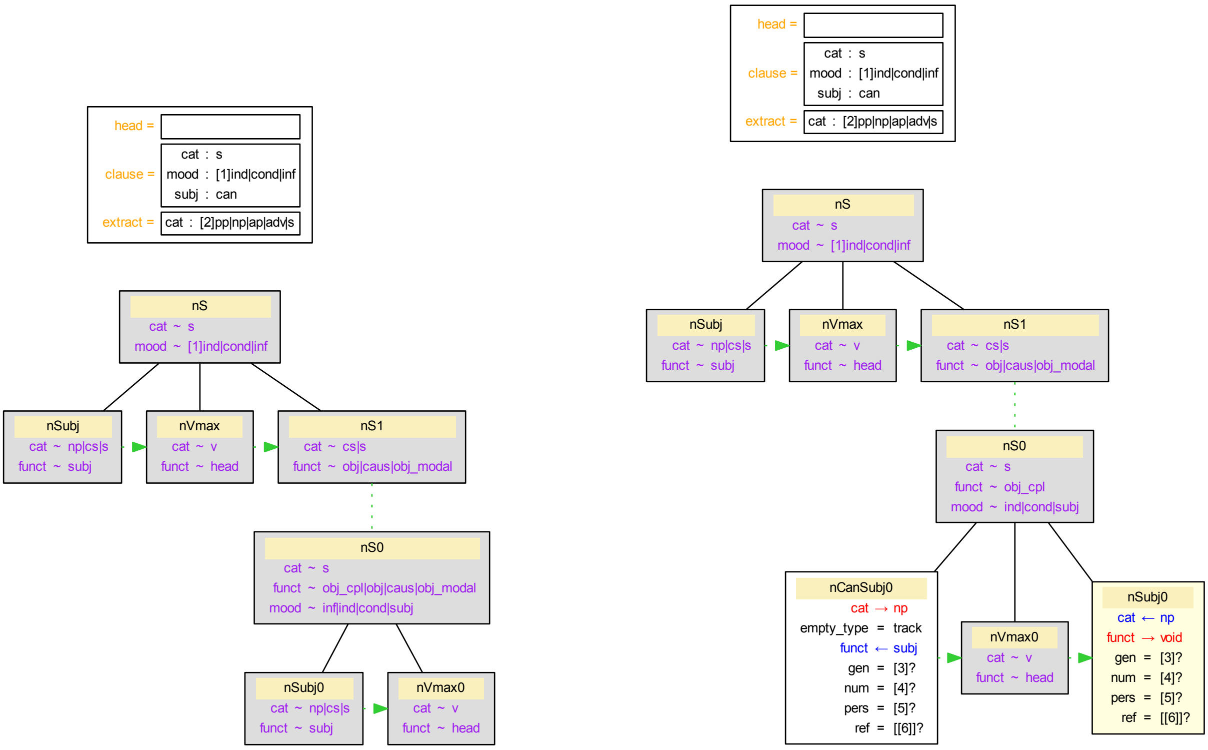

Figure 10.3: The differences between the PTDs defined by the DistantExtractedConstituent class for grammatical words imposing the canonical order subject verb

The examples below illustrate the interaction between these different constraints.

(10.19) *Pourquoi vouloir que Marie descende* □ *?*
why to wish that Marie go down □ ?
Why to wish that Marie go down?

(10.20) *Pourquoi Jean veut-il que descende Marie* □ *?*
why Jean does he wish that go down Marie □ ?
Why does Jean wish that Marie go down?

(10.21) **Pourquoi veut Jean que Marie descende* □ *?*
why wishes Jean that Marie go down □ ?

(10.22) *Que veux-tu que Marie fasse* □ *?*
What do you want that Marie do □ ?
What do you want that Marie do?

(10.23) *Que veux-tu que fasse Marie* □ *?*
What do you want that do Marie □ ?
What do you want that Marie do?

(10.24) **Que veut Jean que Marie fasse* □ *?*
What want Jean that Marie do □ ?

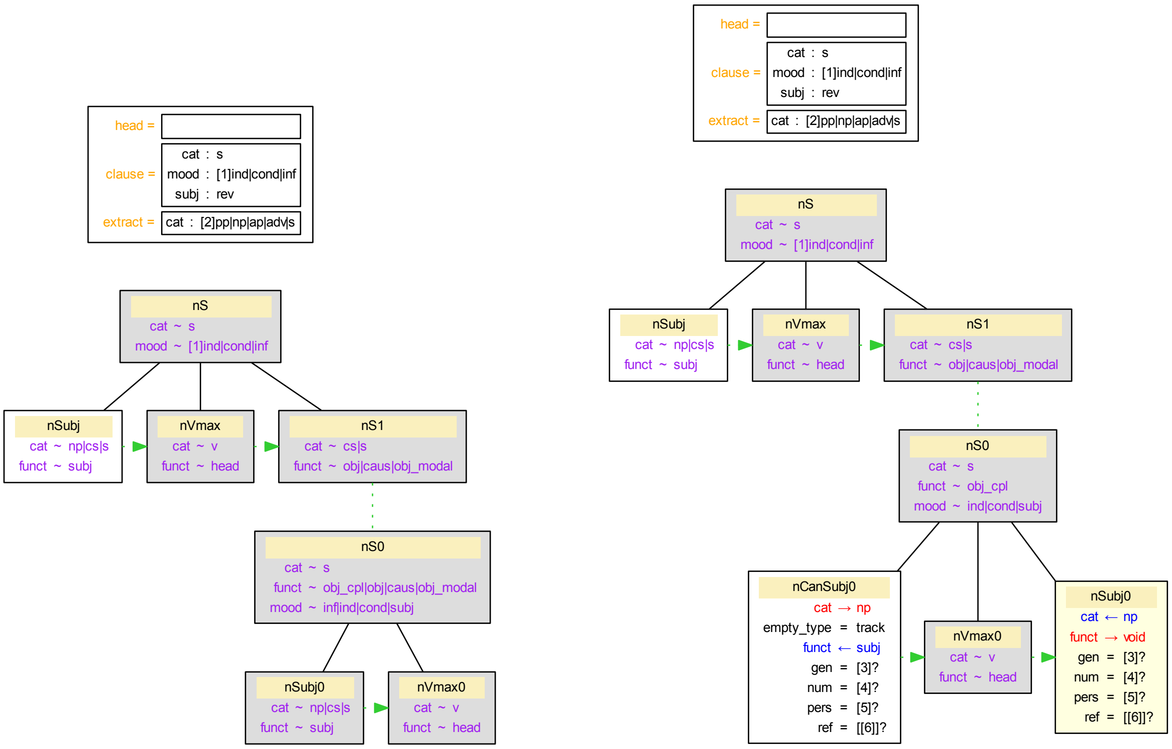

Figure 10.4: The two PTD defined by the DistantExtractedConstituent class for grammatical words imposing the inversion verb subject, without full subject of the *nS*

(10.25) *Que veut faire* □ *Marie ?*
What want do □ Marie ?
What do want Marie to do?

In the sentences above, we use the interrogative adverb *pourquoi*, which imposes the order subject-verb, and the interrogative pronoun *que*, which imposes the inversion verb-subject. These sentences illustrate the fives cases defined by the DistantExtractedComplement class. The class generates five PTD:

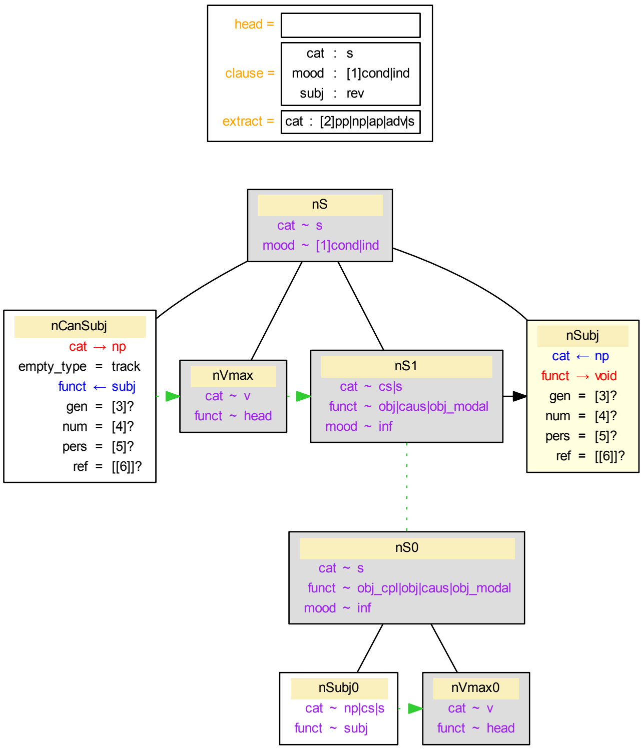

Figure 10.5: The PTD defined by the DistantExtractedConstituent class for grammatical words imposing the inversion verb subject, without full subject of the *nS*

- The two first PTDs, shown in Figure 10.3, correspond to the constraint of the order subject-verb imposed by the grammatical word. This constraint concerns only the clause nS, hence two PTDs corresponding to the two possibilities for the order between the verb and the subject in *nS0*. Sentences (10.19) and (10.20) respect the constraint, but Sentence (10.21) is not grammatical because it violates the canonical order subject verb in *nS*.

- The two following PTDs, shown in Figure 10.4, correspond to the constraint of the order verb-subject imposed by the grammatical word. In the case that *nS0* is embedded in *nS*, the constraint applies in a restrictive way: the subject of *nS* cannot be a full subject, hence Sentence (10.24) is not grammatical; it is not expressed if *nS* is an infinitive or it is a clitic. Sentences (10.22) and (10.23) illustrate this case and they show that the order between verb and subject in *nS0* is free. Figure 10.4 show the corresponding PTDs. The left PTD is used in the parsing of Sentence (10.22), whereas the right PTD is used in the parsing of Sentence (10.23).

- The last PTD, shown in Figure 10.5, correspond to the constraint of the order verb-subject imposed by the grammatical word and to a chain of embedded infinitives starting at *nS1* and ending with *nS0*. It is illustrated with Sentence (10.25), where the chain of infinitives reduces to *faire*. In this case, the expression *veut faire* is considered as a unique verb. If the grammatical word responsible of the extraction imposes verb-subject inversion, this inversion is done in *nS*.

In the following of the module, the difference between the two classes IMMEDIATEEXTRACTEDCOMPLEMENT and DISTANTEXTRACTEDCOMPLEMENT does not matter, so that they gathered in their disjunction EXTRACTEDCOMPLEMENT1.

10.1.2 The different syntactic functions of the extracted constituent

As we will explain later, we consider that extracted constituents can only be complements: direct objects or attributive complements, indirect complements, which can be required of adjunct. The case of subjects will be studied separately. Moreover, they can be complements of verbs, nouns or adjectives. It depends on the grammatical word responsible of the extraction and it is expressed in the lexicon with the feature `gov.cat`, which gives the category of the word governing the extracted constituent: `adj`, `n` or `v`.

Another parameter is the category of the extracted constituent, which is expressed in the lexicon with the feature `extract.cat`. This feature can take the values `ap`, `np` or `s` but the extracted constituent can be introduced by a preposition.

The extracted constituent is a verb complement

The EXTRACTEDVERBCOMPLEMENT class deals with the case that the extracted constituent is a complement of the head verb of *nS0*. it inherits the EXTRACTEDCOMPLEMENT1 class and adds constraints to *nTrace*: it is a daughter of *nS0* which is put after *nVmax0*. Moreover, its possible syntactic function is expressed with the feature `funct = iobj|mod|obj|objpred|subjpred`.

Then, the class is specialized in two sub-classes according to the fact that the complement is required by the verb or adjunct: the EXTRACTEDREQUIREDVERBCOMPLEMENT class and the EXTRACTEDADJUNCTVERBCOMPLEMENT class.

The EXTRACTEDREQUIREDVERBCOMPLEMENT class must distinguish three cases according to the category of the extracted constituent: noun phrase, adjectival phrase or prepositional phrase. It generates three kinds of PTDs. Figure 10.6 shows an example of PTD for the case that the extracted constituent is an adjectival phrase. The trace has a complex structure because it is an adjectival phrase and all adjectival phrases have a subject. In this PTD, the subject is represented by node *nTraceSubj*.

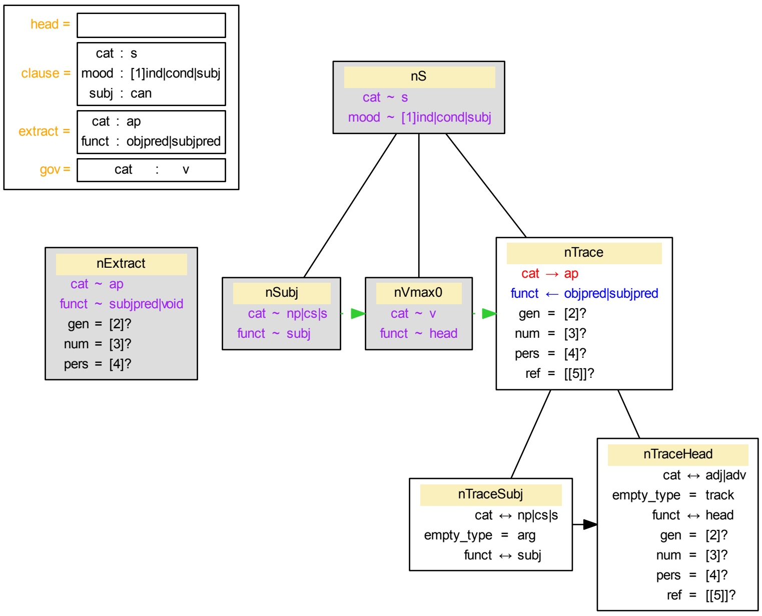

Figure 10.6: A PTD defined by the ExtractedRequiredVerbComplement class for an extracted constituent that is an adjectival phrase

Figure 10.7 shows an example of PTD for the case that the extracted constituent is a prepositional phrase. Here, the specific point is that a positive feature `prep` → `?` provides the preposition that is associated with the grammatical word anchoring the PTD. For instance, when this word is *où*, the feature is `prep` → `loc`. When the grammatical word imposes no preposition, the value of the feature is `void`. Moreover, an empty leaf *nTraceHead* represents the head of the trace as an empty noun phrase. It is used when the verb head of node *nVmax0* imposes a constraint on the category of *nTraceHead*. This constraint must be compatible with the constraint imposed by the grammatical word responsible of the extraction.

The ExtractedAdjunctVerbComplement class corresponds to the case that the extracted constituent is an adjunct of the head verb of *nS0*. In this case, we assume that the extracted constituent is a prepositional phrase. The difference with respect to the corresponding case defined by the ExtractedRequiredVerbComplement class is first the polarities of features attached at node *nTrace*: they are all saturated. Second the value of `funct` is `mod`.

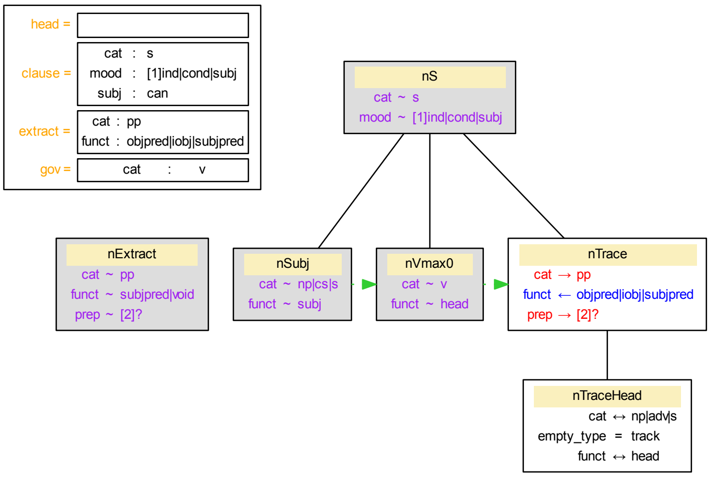

Figure 10.7: A PTD defined by the ExtractedRequiredVerbComplement class for an extracted constituent that is a prepositional phrase

The extracted constituent is a noun complement

The ExtractedNounComplement class applies to the case that the extracted constituent is a noun complement. It inherits the ExtractedComplement1 class and it defines two kinds of PTDs according to the function of the extracted constituent with respect to the noun on which it depends: either it is a modifier or it is a complement required by the noun. Here are examples illustrating both cases.

(10.26)

La	*femme*	*dont*	*Jean*	*a visité*	*la*	*maison*	□	*est*	*une*	*collègue*	*.*
The	woman	whose	Jean	visited	the	house	□	is	a	colleague	.

The woman whose house was visited by Jean is a colleague.

(10.27)

La	*maison*	*dont*	*la*	*construction*	□	*est*	*achevée*	*est*	*celle*	*d'*	*une*	*collègue*	*.*
The	house	of which	the	building	□	is	finished	is	that	of	a	colleague	.

The house the building of which is finished is that of a colleague.

In Sentence (10.26), the relative pronoun *dont* is a modifier of *maison*, whereas in Sentence (10.27) *dont* is a complement required by *construction*.

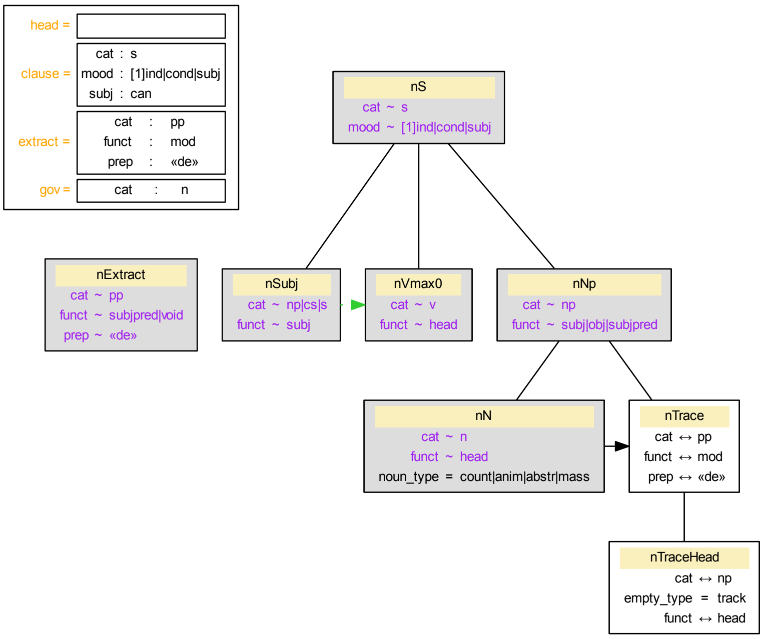

Figure 10.8: PTD defined by the ExtractedNounComplement class for an extracted complement that is a noun modifier

Figure 10.8 shows an example of PTD for an extracted complement that is a noun modifier and Figure 10.9 shows an example of PTD for an extracted complement that is a complement required by a noun. In both cases, node *nN* represents a common noun that has *nTrace* as complement and it builds the noun phrase *nNp* with it. There are restrictions on the function of *nNp*. It may be only the subject, the direct object or the subject predicative complement of the verb head of *nvmax0*. For instance, the following sentence is ungrammatical because the extracted constituent *la femme* is a complement of *mari*, which is not a direct but an indirect object of *parle*.

A common property to both classes is that verb-subject inversion is forbidden. For instance, the following sentence is ungrammatical.

(10.28) **La maison dont est achevée la construction* □ *est celle d'*
The house of which is finished the building □ is that of
une collègue .
a colleague .

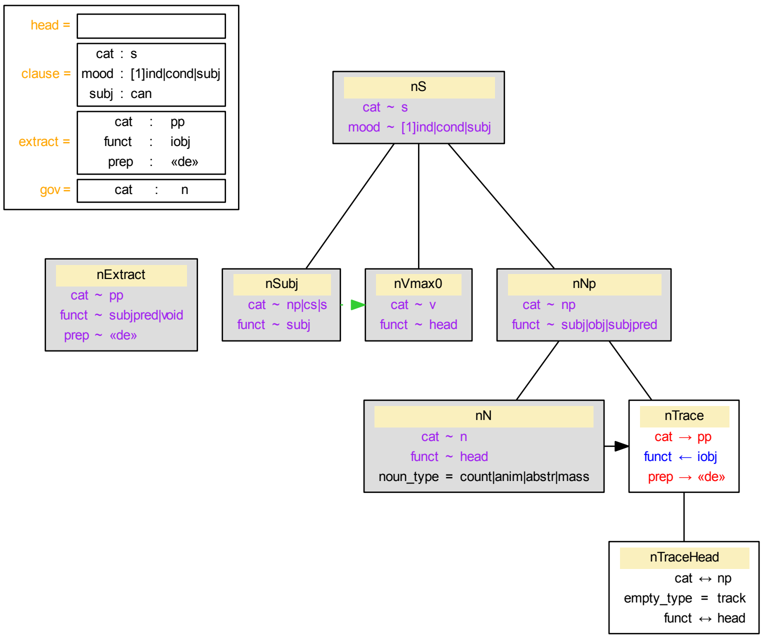

Figure 10.9: PTD defined by the ExtractedNounComplement class for an extracted complement that is a complement required by a noun

This constraint of canonical verb-subject order is added with the feature `clause.subj : can` in the PTD interfaces.

The main difference between the two cases lies in the polarities of the features attached at node *nTrace*. In the case of the modifier, they are saturated. In the case of the complement required by the noun, they are positive or negative.

(10.29) **La femme dont je parle au mari* □ *est une*
The woman whose I am speaking to the husband □ is a
collègue .
colleague .

The extracted constituent is an adjective complement

We consider that the adjective is a subject predicative complement and the extracted complement is a complement required by the adjective. The following example illustrates this case.

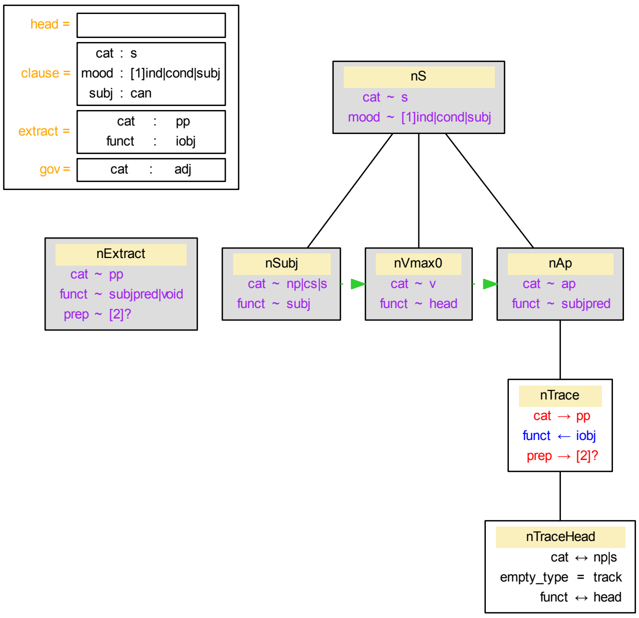

Figure 10.10: PTD defined by the ExtractedAdjectiveComplement class for an extracted complement that is a complement required by an adjective

(10.30) *A quoi doit-on être attentif* □ *?*
What must one be careful about □ ?
What must one be careful about?

The ExtractedAdjectiveComplement class deals with this case. It inherits the ExtractedComplement1 class and it adds node *nAp* which represents the adjectival phrase that has *nTrace* as a daughter. As in the previous case, verb-subject inversion is forbidden, which is also expressed in the PTD interface with the feature `clause.subj : can`.

The four classes ExtractedRequiredVerbComplement, ExtractedAdjunctVerbComplement, ExtractedNounComplement and ExtractedAdjectiveComplement are gathered in the disjunction ExtractedComplement.

10.1.3 Interrogative and relative words attached to subjects

Here are examples of interrogative and relative words attached to subjects (in bold).

(10.31) *L'ingénieur* **qui** *connaît Marie arrive demain .*
the-engineer who knows Marie is-arriving tomorrow .
The engineer who knows Marie is arriving tomorrow.

(10.32) **Lequel** *connaît* *Marie* *?*
who knows Marie ?
Who knows Marie?

(10.33) **Quelle personne** *connaît* *Marie* *?*
which person knows Marie ?
Which person knows Marie?

In these examples, *qui*, *lequel* and *quelle personne* takes the position of the subject of the clause that they introduce. There is no move and therefore there is no need for introducing a trace. Moreover, such words do not allow pied piping.

As a consequence, a specific class NonExtractedSubject represents these words with this function. Figure 10.11 shows the PTDs defined by this class.

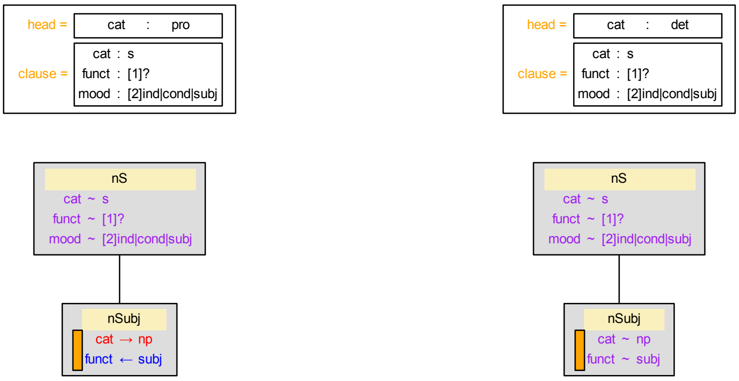

Figure 10.11: The PTDs defined by the NonExtractedSubjectt class

In the PTDs defined by the class, the *nSubj* represents the subject. The difference between the PTDs lies in the polarities attached at this node. The left PTD corresponds to a pronoun which plays the role of a noun phrase subject. Therefore, node *nSubj* carries a positive feature `cat` → `np` and a negative feature `funct` ← `subj`. The right PTD corresponds to a determiner and the head of the noun phrase subject is filled by another word. Therefore, node *nSubj* carries two virtual features `cat` ∼ `np` and `funct` ∼ `subj`.

10.1.4 Pied piping for relative and interrogative words

All relative and interrogative words express extraction, except when they are used as subjects. Extraction gives rise to an unbounded dependency but there is second unbounded dependency that is introduced by some of them in a phenomenon that is called *pied piping*. Pied piping means that the extracted constituent does not necessarily identify with the grammatical word that allows extraction: in this case, the grammatical word represents a determiner or a noun phrase more or less deeply embedded in the extracted constituent, hence a second unbounded dependency between the grammatical word and the head of the extracted constituent.

There are only pronouns *combien*, *qui*, *quoi* and *lequel* pronouns and the determiner *quel* that allow pied piping. Here are examples illustrating the phenomenon. In each example, the extracted constituent is put in square brackets and the relative or interrogative word is displayed in bold.

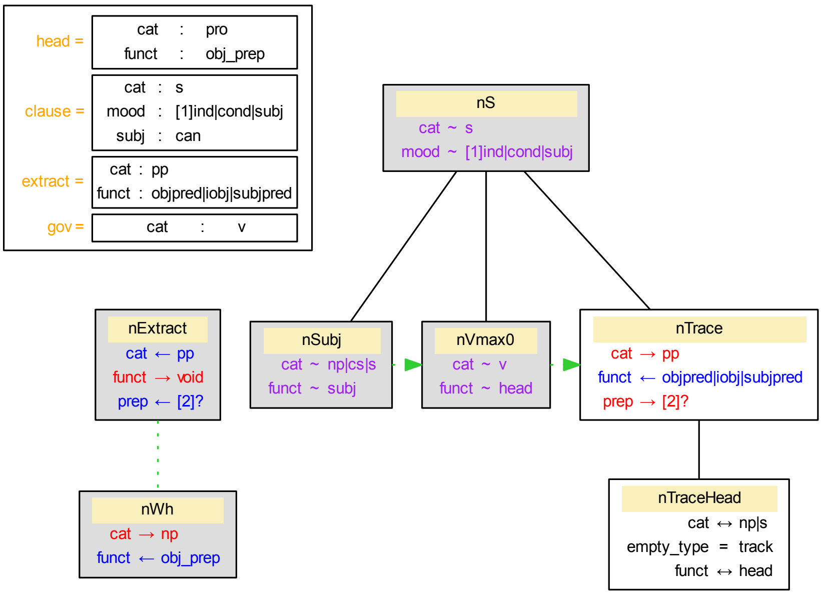

Figure 10.12: A PTD defined by the ExtractedComplementNonHead class

(10.34) *Jean [dans l' entreprise de* **qui***] Marie sait que l' ingénieur*
John in the company of whom Mary knows that the engineer
travaille □ *est malade .*
works is sick .
John, in the company of whom Mary knows the engineer works, is sick.

(10.35) *[Au directeur de* **quelle** *entreprise] Marie veut-elle parler* □ *?*
to-the director of which company Mary wants to-speak ?
Which company does Mary want to speak to the director of?

(10.36) *[Au directeur de* **laquelle***] Marie veut-elle parler* □ *?*
to-the director of which-one Mary wants to-speak ?
Which one does Mary want to speak to the director of?

The ExtractedConstituent is specialized in two sub-classes for relative and interrogative words according to the fact they allow pied piping or not.

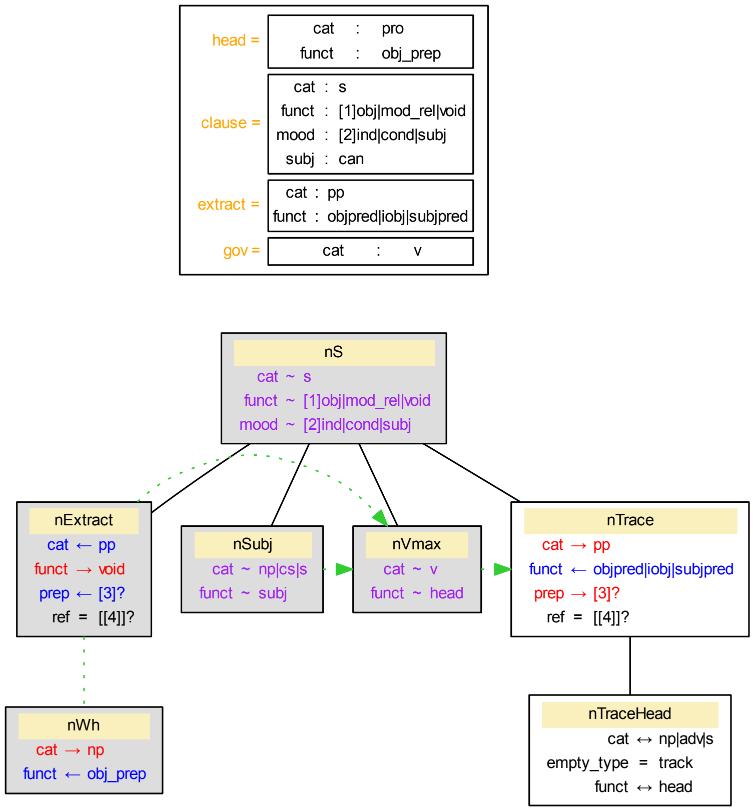

Figure 10.13: The PTD of Figure 10.12 completed by the WhExtractedComplement class

- ExtractedComplementNonHead for relative and interrogative words that are not the head of the extracted constituent in presence of pied piping,
- ExtractedComplementHead for relative and interrogative words that identify to the extracted constituent; as a consequence, it constrains two saturated features `cat ↔ ?` and `funct ↔ void`.

In all cases, the extracted constituent is represented with node *nExtract* and the relative or interrogative word is represented with node *nWh*. The first class introduces an underspecied dominance relation from *nExtract* to *nWh*, whereas the second class identify the two nodes.

Figure 10.12 shows an example of PTD defined by the ExtractedComplementNonHead class. In this example, the extracted constituent is prepositional phrase which is a complement that is required by the head verb of *nS*. The polarized features of node *nExtract* expresses that this node represents an expected prepositional phrase, whereas the polarized features of node *nWh* express that this node is a noun phrase expecting to be the object of a preposition.

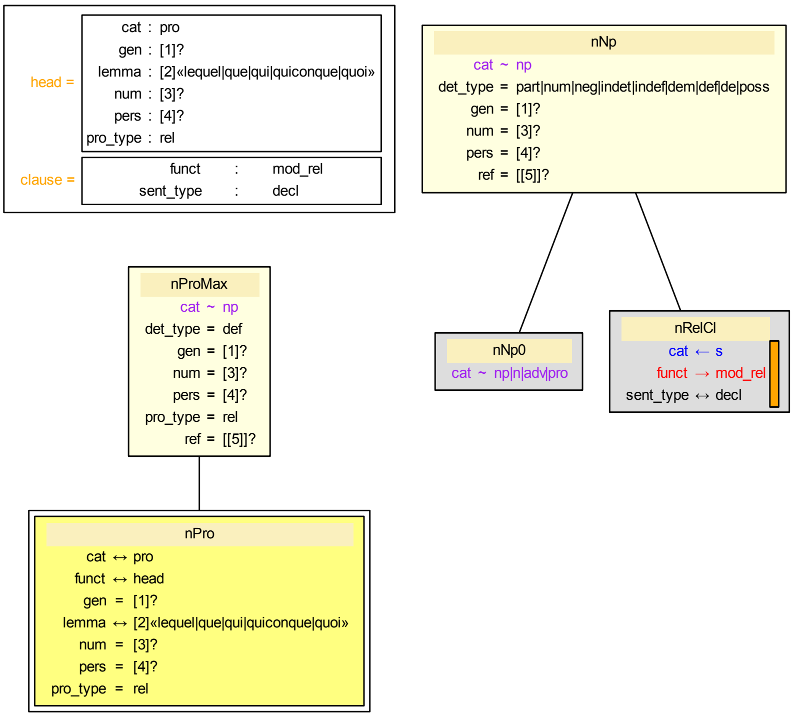

Figure 10.14: A PTD defined by the RelativePronoun class

The ExtractedComplementNonHead and ExtractedComplementHead classes are gathered in their disjunction WhExtractedComplement. Moreover, the class adds properties that are common to relative and interrogative words: node *nExtract* is a daughter of node *nS* and it co-refers with node *nTrace*. Finally, the function of node *nS* is restricted to `mod_rel|obj|void`. Figure 10.13 shows the PTD of Figure 10.12 after addition of new constraints by the WhExtractedComplement class.

Then, the WhExtractedComplement class is specialized in different classes according to the nature of *nS*: interrogative clause or relative clause.

Other classes are dedicated to cleft clauses but they are studied in a next section.

10.2 Relative clauses

The relative clauses are marked with relative pronouns. The RelativePronoun class defines the PTD skeleton that is common to all relative pronouns. It is made up of two PTDs, a first one for pronouns replacing noun phrases (*lequel*, *qui*, *quiconque*,*que*, *quoi*) and a second one for pronouns replacing prepositional phrases (*dont*, *d'où*, *où*). Figure 10.14 shows the first PTD, which is constituted of two parts:

- Node *nPro* represents the node anchoring the relative pronoun and its mother node *nProMax* represents its maximal projection, which is a noun phrase.
- Node *nRelCl* represents the expected relative clause. It will provides it with the function `mod_rel`. The antecedent of the relative pronoun is represented with node *nNp0*, which builds the noun phrase *nNp* with the relative clause.

The relation between the two parts is variable and it is not specified by the RelativePronoun class. Then, this class is specialized according to the function of the relative pronoun.

10.2.1 Standard Complement relative pronouns

The ComplementRelativePronoun class defines the PTDs for complement relative pronouns with antecedents. It is the conjunction of the RelativePronoun and WhExtractedComplement classes. It respectively identifies nodes *nRelCl* and *nProMax* coming from the first class with nodes *nS* and *nWh* coming from the second class.

The class is renamed as the terminal class NP0_PROrel-compl_S1. It defines 72 EPTDs and the sentences below give some examples of their use.

(10.37)	*Je*	*connais*	*la*	*ville*	**d'où**	*Jean*	*est*	*originaire*	*.*
	I	know	the	city	from which	Jean	is	native	.

I know the city of which Jean is native.

(10.38)	*L'*	*ingénieur*	**dont**	*Jean*	*croit*	*que*	*Pierre*	*connaît*	*la*	*décision*
	The	engineer	whose	Jean	believes	that	Pierre	knows	the	decision
	vient	*aujourd'hui*	*.*							
	is coming	today	.							

The engineer whose decision Jean believes that Pierre knows is coming today.

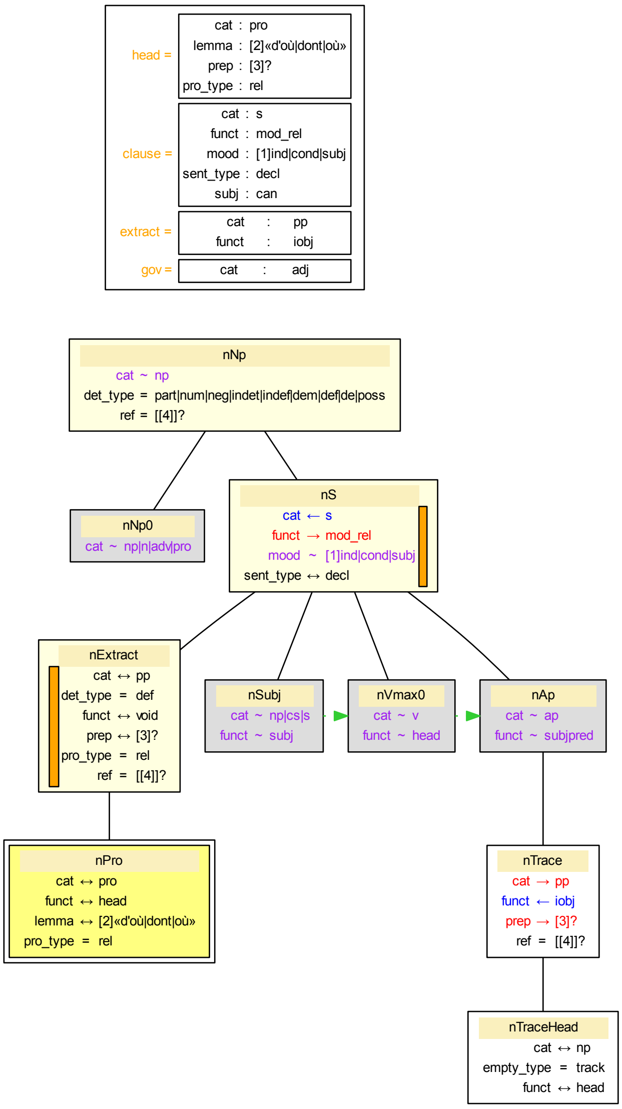

Figure 10.15: EPTD used for the relative pronoun *d'où* in the parsing of Sentence (10.37)

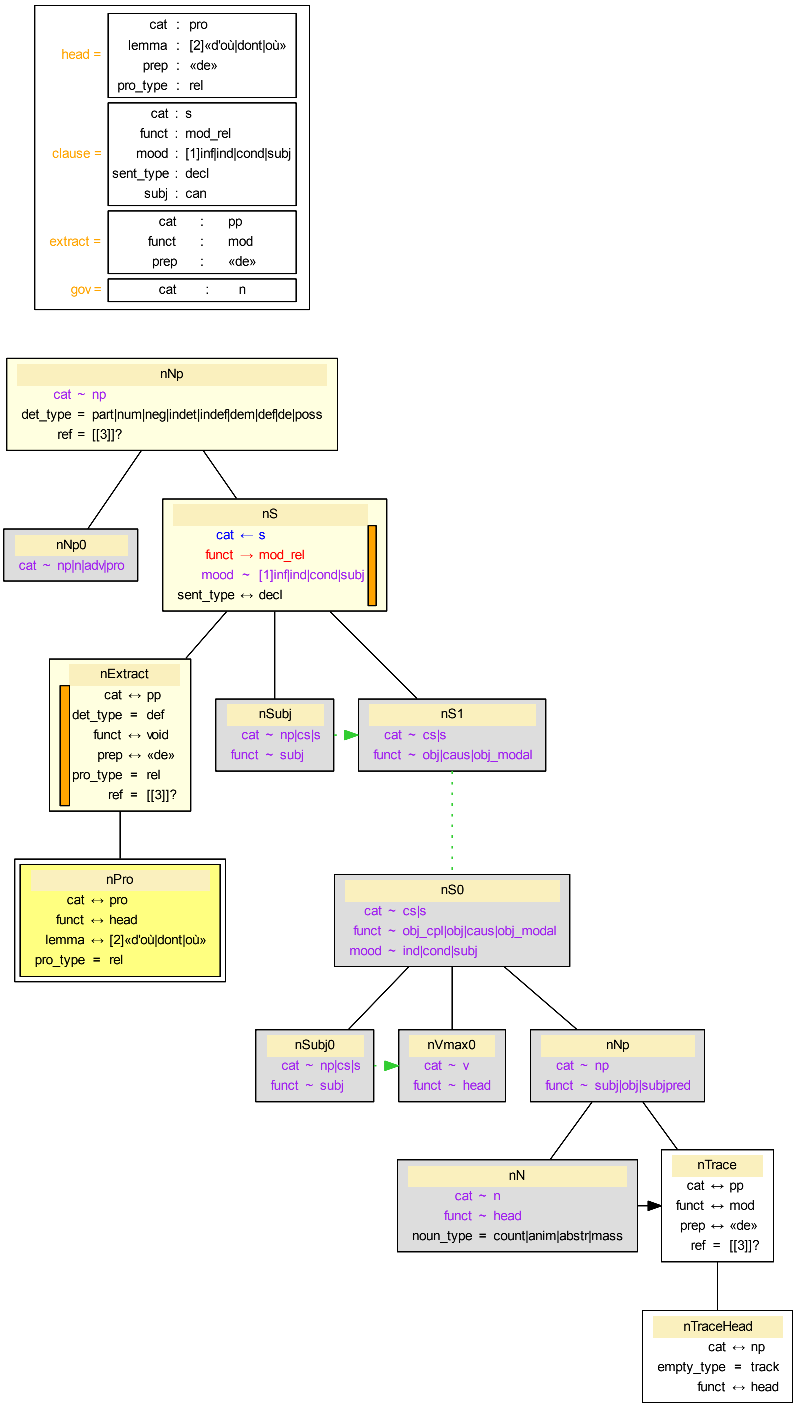

Figure 10.16: EPTD used for the relative pronoun *dont* in the parsing of Sentence (10.38)

(10.39) *Jean connaît ceux à* **qui** *proposer une invitation .*
Jean knows those to whom propose an invitation .

Jean knows those who be proposed an invitation.

(10.40) *Jean connaît Pierre dans l' entreprise* **duquel** *il a rencontré*
Jean knows Pierre in the company whose he met
Marie .
Marie .

Jean knows Pierre in whose company he met Marie.

Figure 10.15 shows the EPTD used for the relative pronoun *d'où* in the parsing of Sentence (10.37). The trace of the relative pronoun, represented with node *nTrace* is a complement required by the adjective *originaire*, which is the head of the adjectival phrase *nAp.* The extracted constituent, represented by node *nExtract*, reduces to the relative pronoun *d'où*, with *la ville* as its antecedent and this antecedent is represented with node *nNp0.*

Figure 10.16 shows the EPTD used for the relative pronoun *dont* in the parsing of Sentence (10.38). The trace of the relative pronoun, represented with node *nTrace* is an adjunct of the noun *décision*, which is the head of the noun phrase *nNp.* The clause *Pierre connaît la décision*, represented by node *nS0*, which is the location of the trace, is imbedded in the relative clause *nS.* In this example, nodes *nS0* and *nS1* will be merged in the parsing process.

What is interesting in Example (10.39) is that the relative clause is in the infinitive mood. The relative pronoun *qui* is associated with the EPTD of Figure 10.17. An original feature of the EPTD with respect to the ones that have just be described is that the extracted constituent do not coincide with relative pronoun: there is an underspecified dominance relation from node *nExtract* to node *nWh*, which can give rise to pied piping. A more significant example of pied piping is given by Sentence (10.40), where the extracted constituent is *dans l'entreprise duquel* and the relative pronoun *sequel* is embedded in this constituent.

The extracted constituent, represented by node *nExtract*, reduces to the relative pronoun *dont*, with *l'ingénieur* as its antecedent and this antecedent is represented with node *nNp0.*

10.2.2 Relative pronouns without antecedent

The relative pronouns *où*, *qui* and *quoi* can be used without antecedent. We will study the case of *qui* separately and we deal here with complement relative pronouns. Here are examples that illustrate this use.

(10.41) *Marie sait de* **quoi** *Jean est malade .*
Marie knows from what Jean is ill .

Marie knows from what Jean is ill.

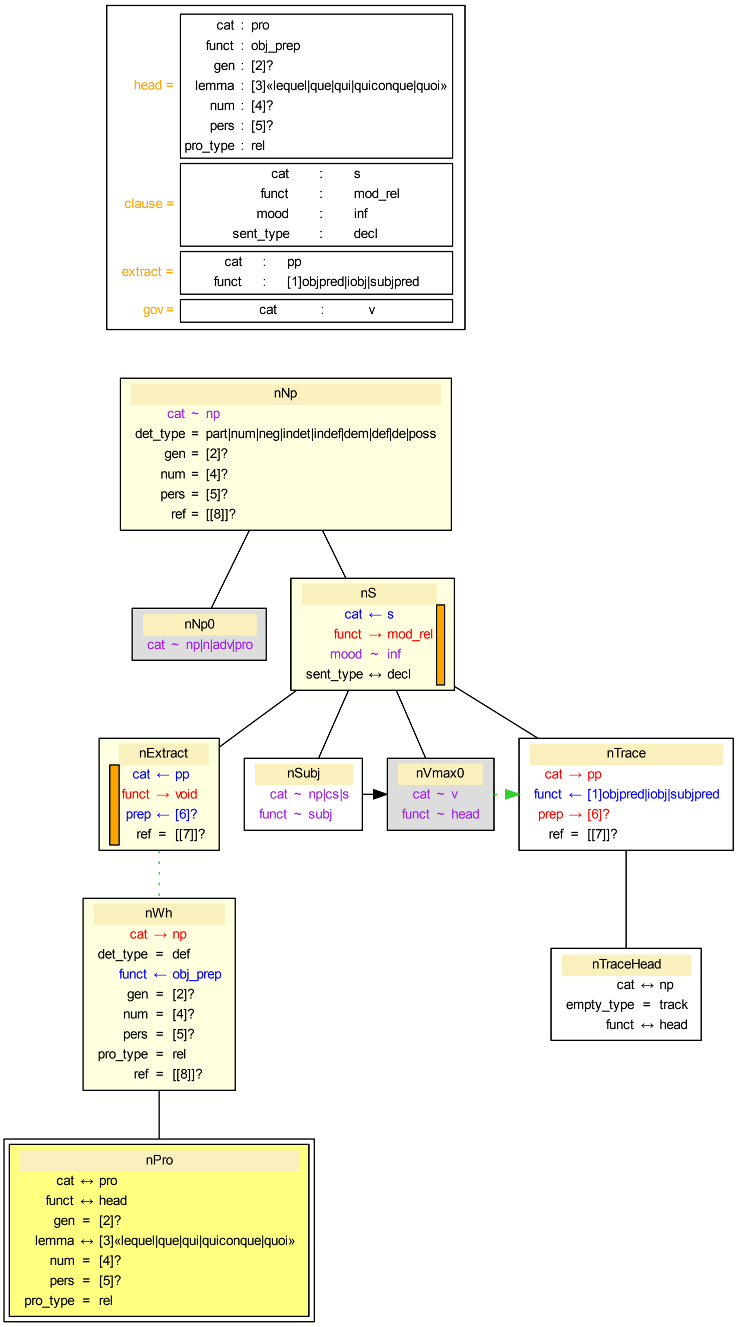

Figure 10.17: EPTD used for the relative pronoun *qui* in the parsing of Sentence (10.39)

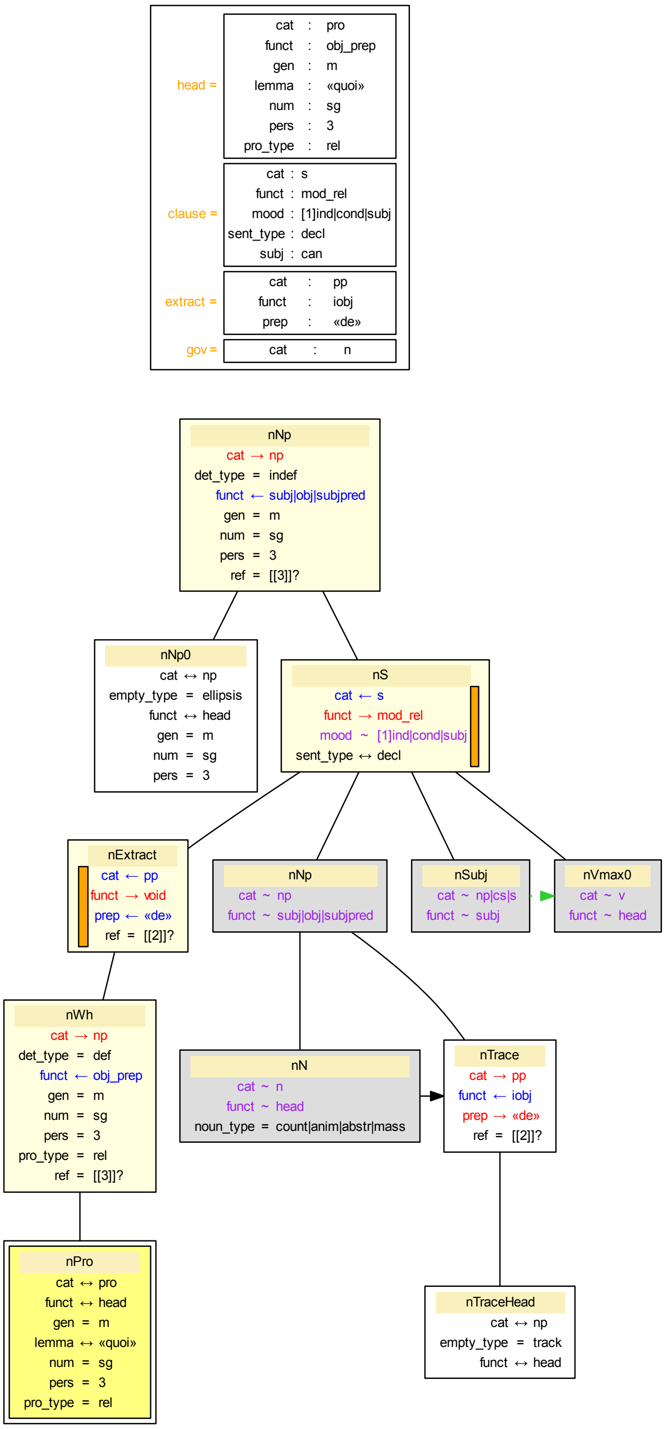

Figure 10.18: EPTD used for the relative pronoun *quoi* in the parsing of Sentence (10.41)

(10.42) *j' irai* **où** *vous voudrez .*
I will go where you want .
I will go where you want.

(10.43) *Jean dort* **où** *il a pu trouver une place .*
Jean is sleeping where he could find a place .
Jean is sleeping where he could have found a place.

The ComplementRelativePronounWithoutAntecedent class gives the skeleton of all EPTDs associated with complement relative pronouns without antecedent. It inherits the ComplementRelativePronoun class and adds some properties to node *nNp0*, which represents the antecedent of the relative pronoun. This node is empty and it has saturated features `cat` ↔ `np` and `funct` ↔ `head`.

Then, the class is specialized in two classes:

- PROrel-quoi_S1 for the relative pronoun *quoi*. This class defines 28 EPTDs among which that one used in the parsing of Sentence (10.41). This EPTD is shown in Figure 10.18.
- PROrel-ou_S1 for the relative pronoun *où*. This class defines 56 EPTDs among which those used in the parsing of Sentences (10.42) and (10.43). These EPTDs are shown in Figure 10.19.

10.2.3 Subject relative pronouns

The subject relative pronouns *qui*, *lequel* and *quiconque* differ from the complement relative pronouns in the fact that they do not give rise to extraction: the subject pronouns are in situ. Then, we must distinguish three cases according to the existence or not of an antecedent and in case of absence of antecedent, according to the function of the relative pronoun in the main clause.

Here are examples illustrating the different cases.

(10.44) *Jean connaît celui* **qui** *vient .*
Jean knows this who is coming .
Jean knows this who is coming.

(10.45) **Quiconque** *connaît Marie peut le dire .*
Anybody knows Marie can it say .
Anybody who knows Marie can say it.

(10.46) *Jean l' entend* **qui** *vient .*
Jean him **is hearing** who is coming .
Jean is hearing who is coming.

(10.47) *C' est lui* **qui** *vient .*
That is him who is coming .
He is coming.

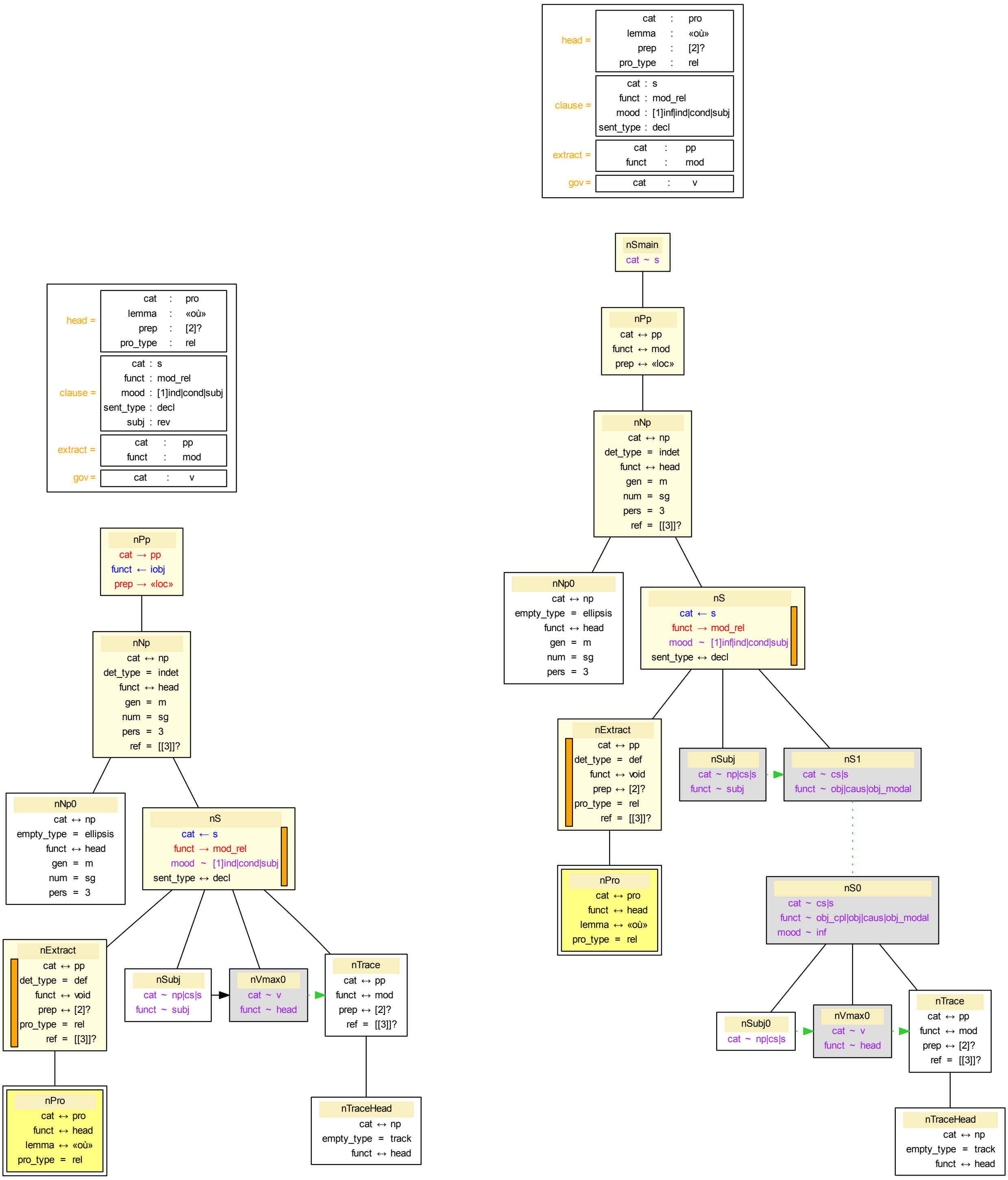

Figure 10.19: EPTDs used for the relative pronoun *où* in the parsing of Sentences (10.42) and (10.43)

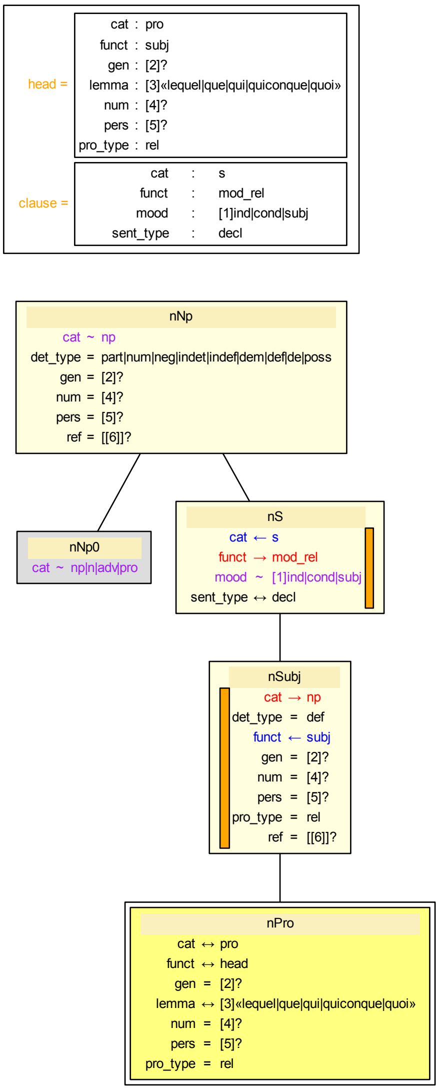

Figure 10.20: PTD defined by the SubjectRelativePronoun class and used for subject relative pronouns

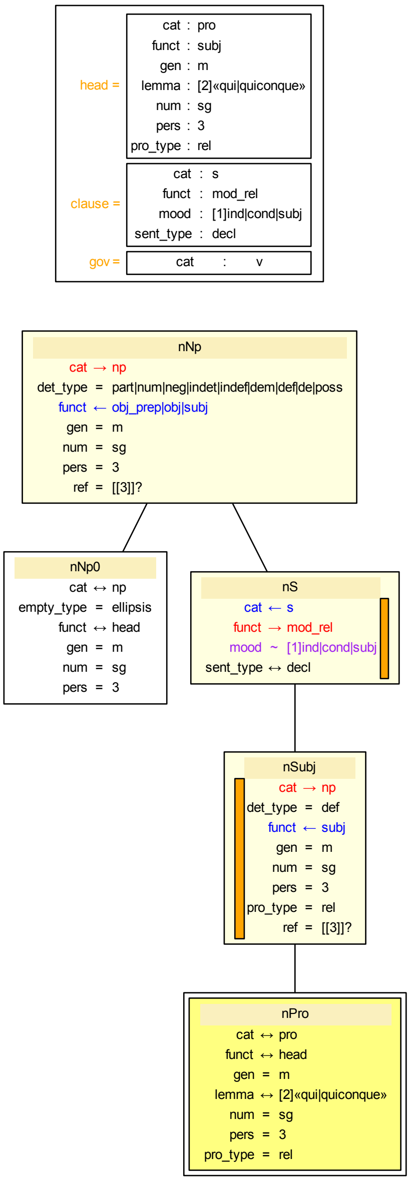

Figure 10.21: EPTD defined by the PROrel-subj_S1np class for subject relative pronouns without antecedent

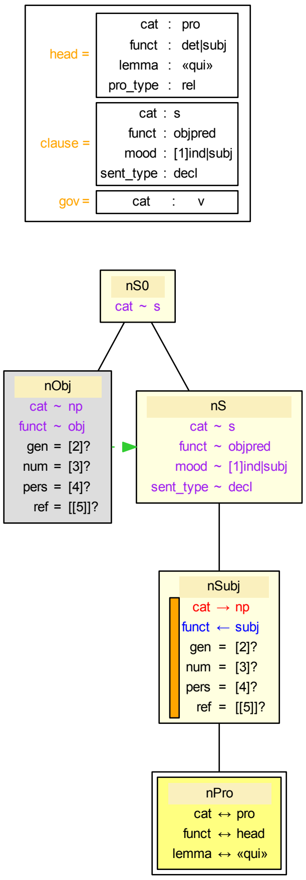

Figure 10.22: EPTD defined by the PROrel-subj_S1objpred class for subject relative pronouns introducing clauses with a function of object. predicative complement

In Sentence (10.44), the relative pronoun *qui* is used in a standard way with an antecedent, the demonstrative pronoun *celui*. In other sentences, the relative pronouns have no antecedent or the relative clauses have specific syntactic functions. In Sentence (10.45), the relative pronoun *quiconque* is subject of *connaît* in the relative clause and subject of *peut* in the main clause. In Sentence (10.46), the relative pronoun *qui* is subject of *vient* in the relative clause and this clause is object predicative complement of the verb *entend*. Sentence (10.47) is a cleft clause and the relative pronoun *qui* is in anaphoric relation with the demonstrative pronoun *c'*.

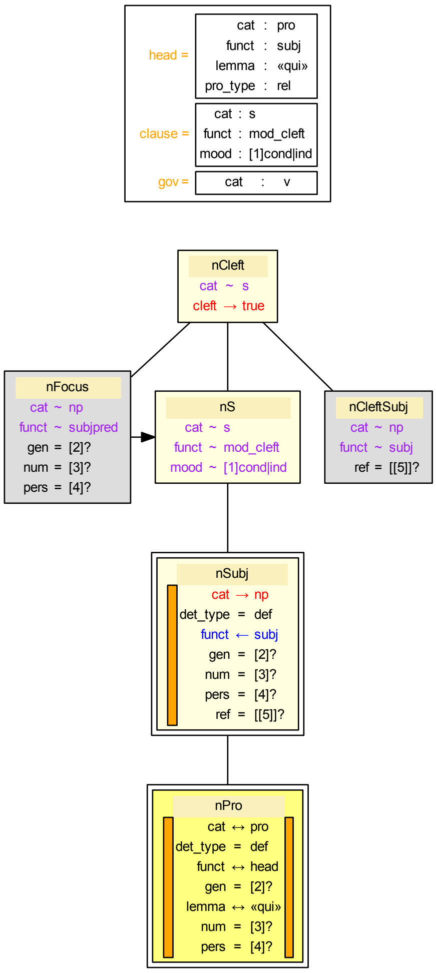

Figure 10.23: EPTD defined by the PROrel-subj_S1cleft class for subject relative pronouns in the context of a cleft clause.

A basic class SubjectRelativePronoun take all these cases into account. It is the conjunction of the classes NonExtractedSubject and RelativePronoun and it defines the PTD shown in Figure 10.20. Node *nPro* is attached at the relative pronoun and it is the head of the noun phrase *nSubj*, which is the subject of the relative clause *nS* and which is located at the beginning of the clause. The antecedent of the relative pronoun is represented by node *nNp0*. Finally, node *nNp* represents the noun phrase constituted of the antecedent modified by the relative clause.

Then, the SubjectRelativePronoun class is specialized in two classes according to the presence or not of an antecedent:

- NP0_PROrel-subj_S1, when the relative pronoun has an antecedent (Sentence (10.44)); the class comes from a renaming of the SubjectRelativePronoun class;
- PROrel-subj_S1np, when the relative pronoun has no antecedent and a syntactic function in the main clause besides its function of subject in the relative clause (Sentence (10.45)); Figure 10.21 shows the EPTD defined by this class; the class adds two polarized features `cat` → `np` and `funct` ← `obj_prep|obj|subl`, which means that the relative pronoun can have a restricted syntactic function in the main clause; node *nNp0* is empty because of the antecedent is considered as elided.

Sentence (10.46) illustrates the case that the relative pronoun *qui* has an antecedent, the personal pronoun *l'*, and the relative clause is an object predicative complement of the verb *extend*. The PROrel-subj_S1objpred class models this behavior and it defines the EPTD shown in Figure 10.22. Node *nObj* represents the antecedent of the relative pronoun *qui*, which has a function of object in the main sentence. The relative clause *nS* is object predicative complement of the head verb of the main clause. It is expressed with the polarized features `cat` ← `s` and `funct` → `objpred`.

There is a last specific use of the subject relative pronoun *qui*, in the context of a cleft clause. Sentence (10.47) illustrates this case and the PROrel-subj_S1cleft class defines the corresponding EPTD, which is shown by Figure 10.23. The relative pronoun co-refers with the subject of the cleft clause, *c'* in our example, which is expressed with feature `ref`. The focus of the cleft clause is represented with node *nFocus* and it is subject predicative complement of the verb *est*, head of the cleft clause.

10.3 Interrogative clauses

In written French language, interrogation is marked with different means: punctuation, grammatical words, subject inversion. Here are examples of such means for the case of partial interrogation with extracted complements. When the interrogation concerns subjects, there is no extraction and this case will be studied separately.

(10.48) *Où* *Jean* *va-t-il* *?*
where Jean is-going ?
where is going Jean?

(10.49) *Où* *va* *Jean* *?*
where is-going Jean ?
where is going Jean?

(10.50) *Où* *va-t-il* *?*
where is-going-he ?
where is he going?

(10.51) *Où* *aller* *?*
where to go ?
where to go?

(10.52) _que fait Jean ?_
what is doing Jean ?
what is Jean doing?

(10.53) *_que Jean fait-il ?_
what Jean is-doing ?
what is Jean doing?

(10.54) _que fait-il ?_
what is-doing-he ?
what is he doing?

(10.55) _pourquoi Jean vient-il ?_
Why Jean is-coming ?
why is Jean coming?

(10.56) _Marie demande pourquoi Jean vient ._
Marie asks why Jean is-coming .
Marie asks why Jean is coming.

(10.57) _Marie demande où va Jean ._
Marie asks where is going Jean .
Marie asks where Jean is going

(10.58) _Marie demande où Jean va ._
Marie asks where Jean is going .
Marie asks where Jean is going

(10.59) *_Marie demande où va-t-il ._
Marie asks where is going he .

(10.60) _Marie demande où aller ._
Marie asks where to go .
Marie asks where to go.

In each of these examples, several marks contribute to the interrogative type of a clause. When interrogation is direct (Examples (10.48) to (10.55)), there is always a question mark at the end of the sentence. Moreover, when interrogation is partial, as in all examples, an interrogative word is used and there is subtle game between verb-subject inversion and the use of a subject clitic put after the verb.

The interrogative pronoun *où* imposes no order between the verb and the subject but when the sentence is finite, it imposes either inversion verb-subject (Examples (10.49) and (10.50)) or the addition of a subject clitic after the verb (Example (10.48)). When the sentence is an infinitive, the alternative does not matter because there is no apparent subject (Example (10.51)).

The interrogative pronoun *que* imposes verb-subject inversion (Examples (10.52) and (10.54)). Therefore, Example (10.53) is ungrammatical. This is the contrary for *pourquoi*, which imposes the canonical order subject-verb.

When interrogation is indirect (Examples (10.56) to (10.60)), there is no question mark at the end but the verb that requires an interrogative clause, *demander* in our examples, can be considered as marking an interrogation. Moreover, an interrogative word represents the questioned expression, and the order verb-subject is more free than for direct interrogation (Examples (10.57) and (10.58)). Nevertheless, the use of a subject clitic after the verb is forbidden (Example (10.59)), because it is reserved for direct interrogation.

In the IG formalism, a way of modeling this complex interaction is to use a polarized feature `sent_type` with the value `inter` attached at the node representing the interrogative clause. Every case is modeled by a specific class, which inherits the WhExtractedComplement class. There are four classes corresponding to four cases:

- the FiniteDirectInterrogativeWithCanonicalSubject class for finite direct interrogative clauses with the canonical order subject-verb; this case is illustrated with Sentences (10.48) and (10.55); the negative feature `sent_type` ← `inter` is brought by the question mark and the positive feature `sent_type` → `inter` by the subject clitic put after the verb; the PTD defined by the FiniteDirectInterrogativeWithCanonicalSubject class and attached at the interrogative word only verifies that *nS* is interrogative with the virtual feature `sent_type` ∼ `inter` and it is finite with the virtual feature `mood` ∼ `cond | ind`; since the interrogation is direct, a virtual feature `funct` ∼ `void` is attached at node *nS*;

- the FiniteDirectInterrogativeWithInvertedSubject class for finite direct interrogative clauses with the inversion verb-subject; the negative feature `sent_type` ← `inter` is brought by the question mark and for the positive feature `sent_type` → `inter`, we must distinguish two sub-cases:
  - the subject is a full noun phrase and the positive feature is brought by the interrogative word (Sentences (10.49) and (10.52));
  - the subject is a clitic and the positive feature `sent_type` → `inter` is brought by the clitic put after the verb (Sentences (10.50) and (10.54));

  The other features brought by the class are the same as in the previous case;

- the InfinitiveInterrogative class for infinitive interrogative clauses; the negative feature `sent_type` ← `inter` is brought by the question mark when the interrogation is direct and by the verb on which the interrogative clause depends if the interrogation is indirect; the positive feature `sent_type` → `inter` is brought by by the interrogative word (Sentences (10.51) and (10.60));

- the FiniteIndirectInterrogative class for finite indirect interrogative clauses; the negative feature `sent_type` ← `inter` is brought by the verb on which the interrogative clause depends and the positive feature `sent_type` → `inter` is brought by the interrogative word (Sentences (10.57), (10.58) and (10.59)).

The *est-ce que* word does not fail in these four cases because it is considered as a complex phrase introducing an interrogative cleft clause. That is why it will be studied in Section 10.4.

The four classes FiniteDirectInterrogativeWithCanonicalSubject, FiniteDirectInterrogativeWithInvertedSubject, InfinitiveInterrogative and FiniteIndirectInterrogative are gathered in their disjunction InterrogativeWithExtractedComplement. Then, this class is specialized according to the nature of the grammatical word, pronoun, adverb or determiner.

10.3.1 Interrogative Pronouns

Here are examples that illustrate the different uses of interrogative pronouns.

(10.61) **Où** *Jean va-t-il ?*
where Jean is-going ?
where is going Jean?

(10.62) **que** *fait Jean ?*
what is doing Jean ?
what is Jean doing?

(10.63) *Jean ne sait pas* **quoi** *faire .*
Jean knows not what to do .
Jean does not know what to do.

(10.64) **Quel** *est cet homme ?*
Who is this man ?
Who is this man?

(10.65) **Combien** *viennent aujourd'hui ?*
How many are coming today ?
How many people are coming today?

The ComplementInterrogativePronoun class defines 201 PTDs giving all possible contexts for complement interrogative pronouns. It inherits the InterrogativeWithExtractedComplement class. The class introduces node *nPro*, which represents the interrogative pronoun and is the daughter of node *nWh* coming from the InterrogativeWithExtractedComplement class.

The class is specialized in four sub-classes:

- PROinter-indircompl_S1 for standard interrogative pronouns that allow the extracted constituent to be an indirect complement. Figure 10.24 shows an EPTD defined by this class and used in the parsing of Sentence (10.61).

- PROInter-dircompl_S1 for standard interrogative pronouns for which the extracted constituent is a direct complement. Figure 10.25 shows an EPTD defined by this class and used in the parsing of Sentence (10.62). When the interrogative pronoun requires subject inversion, we consider that the pronoun is a complement of the head verb of the interrogative verb.

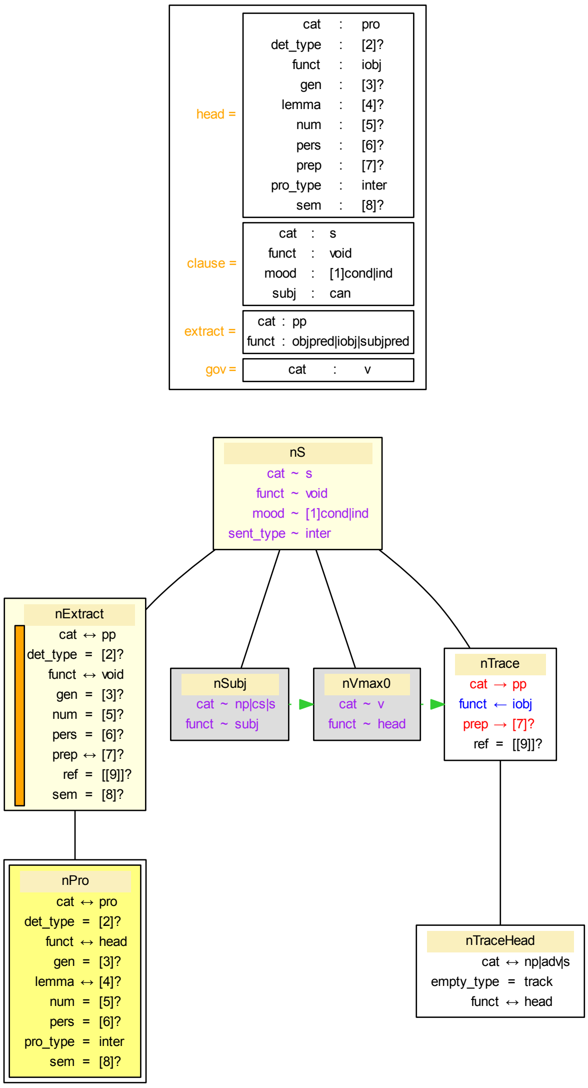

Figure 10.24: EPTD defined by the PROInter-indircompl_S1 class and used in the parsing of Sentence (10.61).

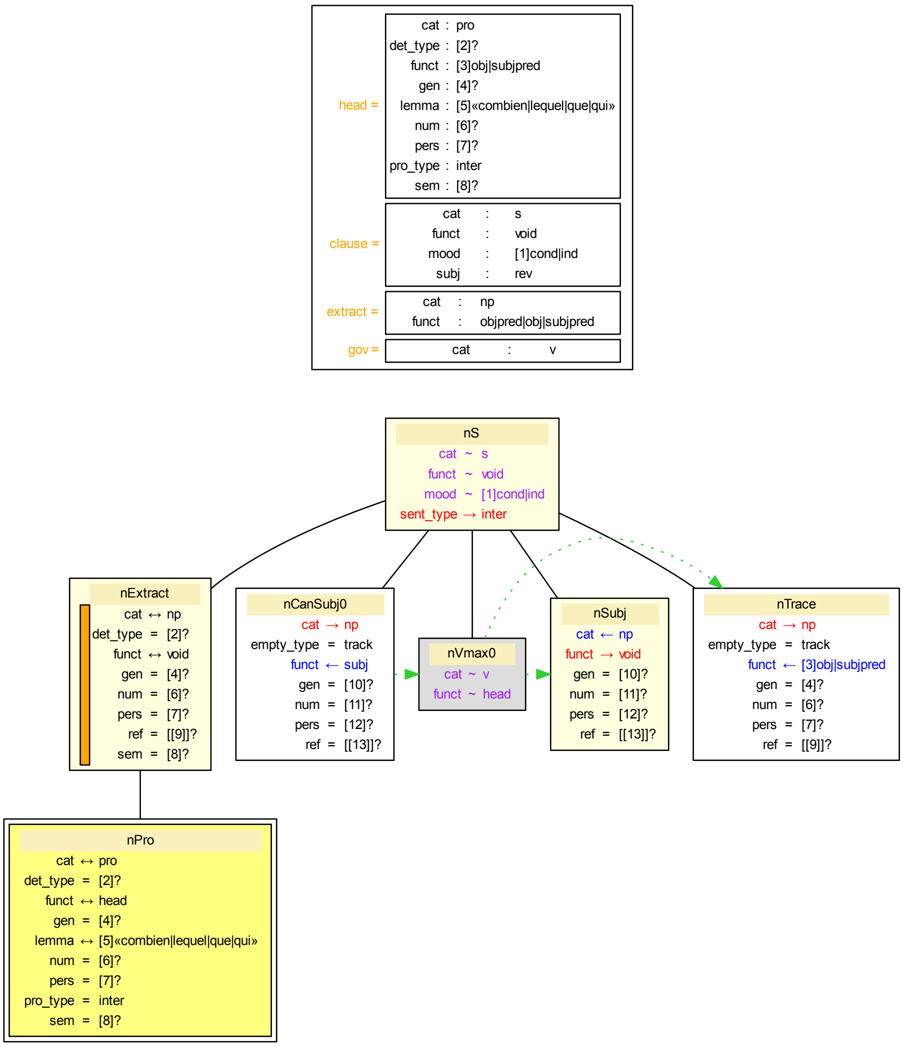

Figure 10.25: EPTD defined by the PROINTER-DIRCOMPL_S1 class and used in the parsing of Sentence (10.62).

- PROQUOI_S1 for the interrogative pronoun *quoi* used as a direct complement, This case is specific because the interrogative clause and all possible clauses that are embedded in it must be infinitives. Figure 10.26 shows an EPTD defined by this class and used in the parsing of Sentence (10.63).
- PROQUEL_S1 for the interrogative pronoun *quel* used as a subject predicative complement. This case is not included in the PROINTER-DIRCOMPL_S1 class because

the pronoun *quel* imposes specific constraints: the mood of the interrogative clause is finite and if the trace in embedded in an object clause, this clause must be infinitive; moreover, the subject must have the third person and if it is a clitic pronoun, the interrogation must be direct. Figure 10.27 shows an EPTD defined by this class and used in the parsing of Sentence (10.64).

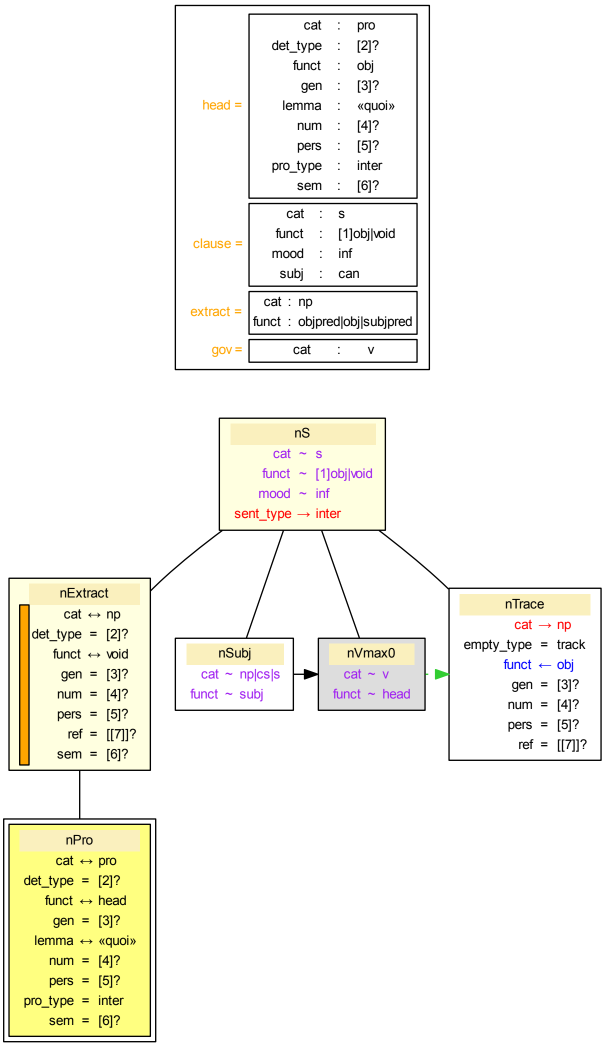

Figure 10.26: EPTD defined by the PROquoi_S1 class and used in the parsing of Sentence (10.63).

(10.66)	**Combien**	*de*	*personnes*	*viennent*	*aujourd'hui*	*?*
	How many	of	persons	are coming	today	?

How many persons are coming today?

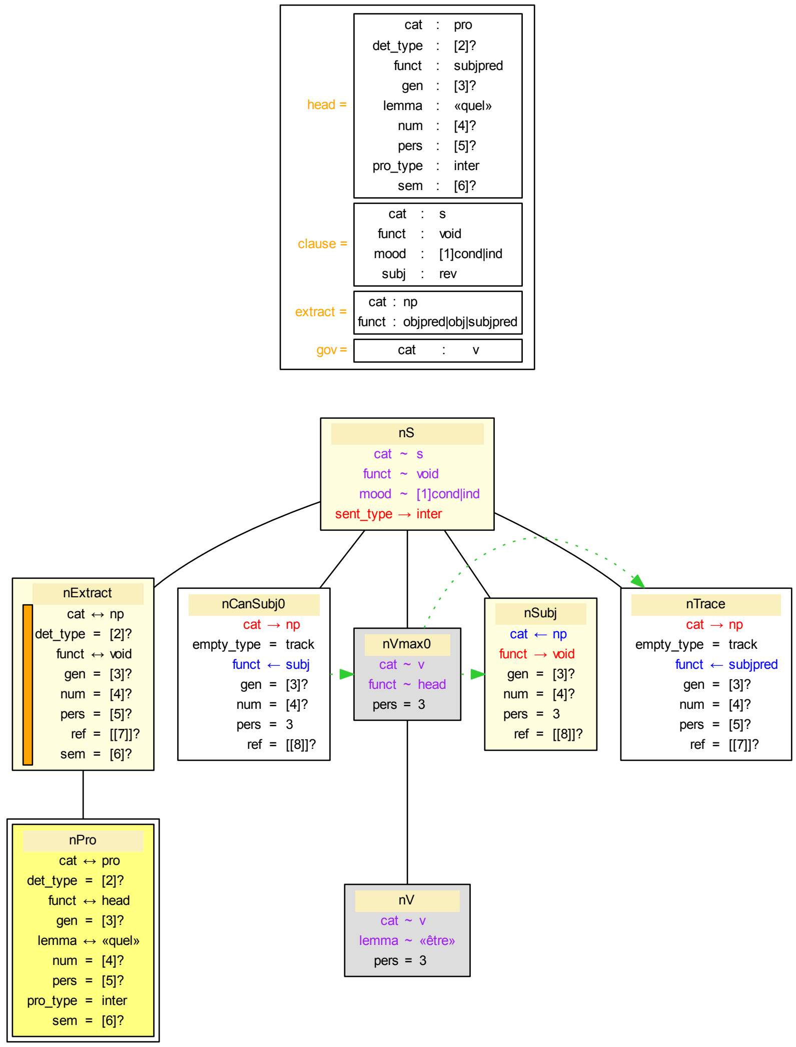

Figure 10.27: EPTD defined by the PROQUEL_S1 class and used in the parsing of Sentence (10.64).

Interrogative pronouns that have a function of subject are studied separately because they are in situ: they do not give rise to extraction. Exemples 10.65 and 10.66 illustrate it: *combien*, which are the subject of *viennent*, are located at the canonical position of any subject, before the verb. A particular class, PROINTER-SUBJ_S1, defines the EPTD attached at such pronouns and used in the parsing of Sentence (10.65). This EPTD is shown on the left of Figure 10.28.

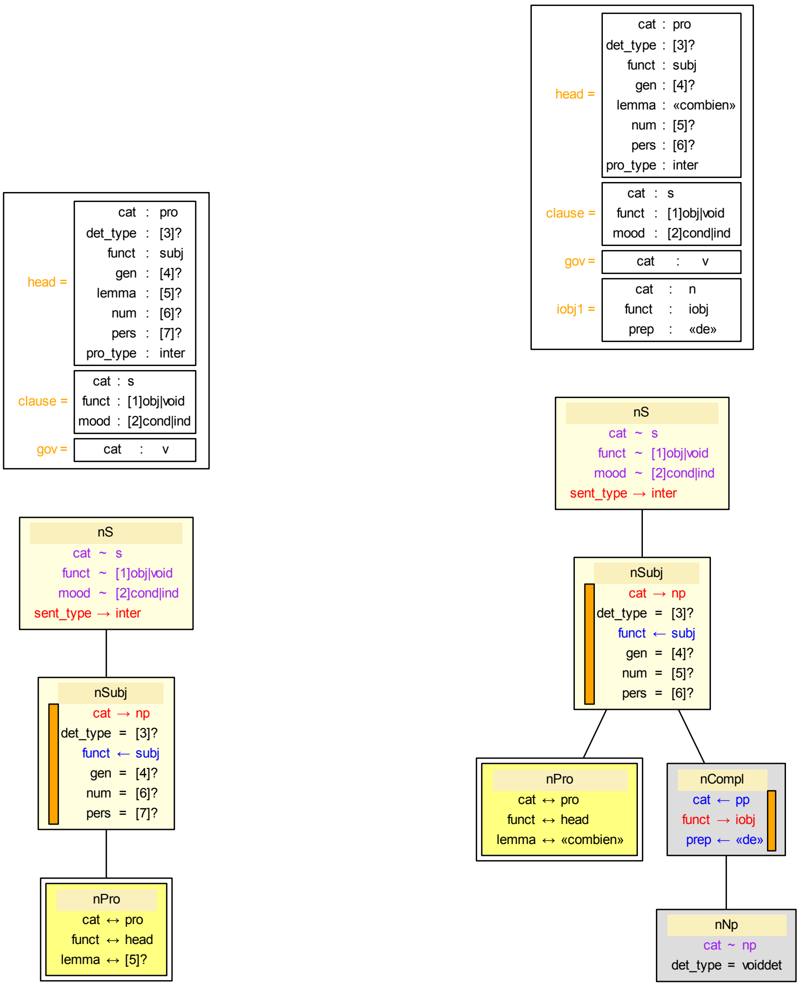

Figure 10.28: The EPTD defined for subject interrogative pronouns without and with a complement.

Some interrogative pronouns, like nouns, require complements, which are obligatory or optional. Sentence (10.53) gives an example with the interrogative pronouns *combien*, which takes *de personnes* as a complement. The PROINTER_DEN1_S2 class defines EPTDs attached at such pronouns. The EPTD used in the parsing of Sentence (10.53) is shown on the right of Figure 10.28.

10.3.2 Interrogative Adverbs

There are three interrogative adverbs, *comment*, *pourquoi* and *quand*. Examples (10.55) and (10.56) illustrate the use of *pourquoi*. Here are examples with *comment*.

(10.67) **Comment** *est* *Jean* *aujourd'hui* *?*
How is Jean today ?
How does fill Jean today?

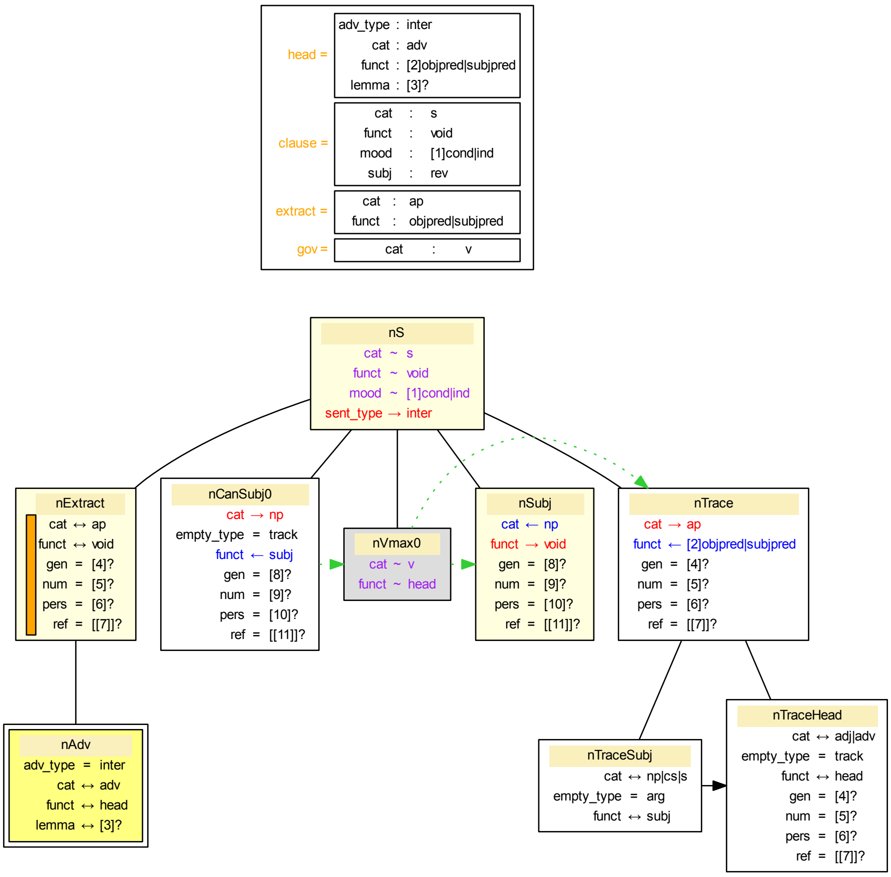

Figure 10.29: An EPTD defined by the ADVINTER_S1 class and used in the parsing of Sentence (10.67).

(10.68) *Marie* *demande* **comment** *va* *Jean* *.*
Marie asks how feels Jean .
Marie asks how Jean feels.

(10.69) **Comment** *aller* *à* *Paris* *?*
How to go to Paris ?
How to go to Paris?

A unique class, ADVINTER_S1, defines the EPTDs associated with interrogative adverbs. It defines 63 EPTDs. Figures 10.29, 10.30 and 10.31 show the EPTDs used for the interrogative adverbs in the parsing of the respective Sentences (10.67), (10.68) and (10.69).

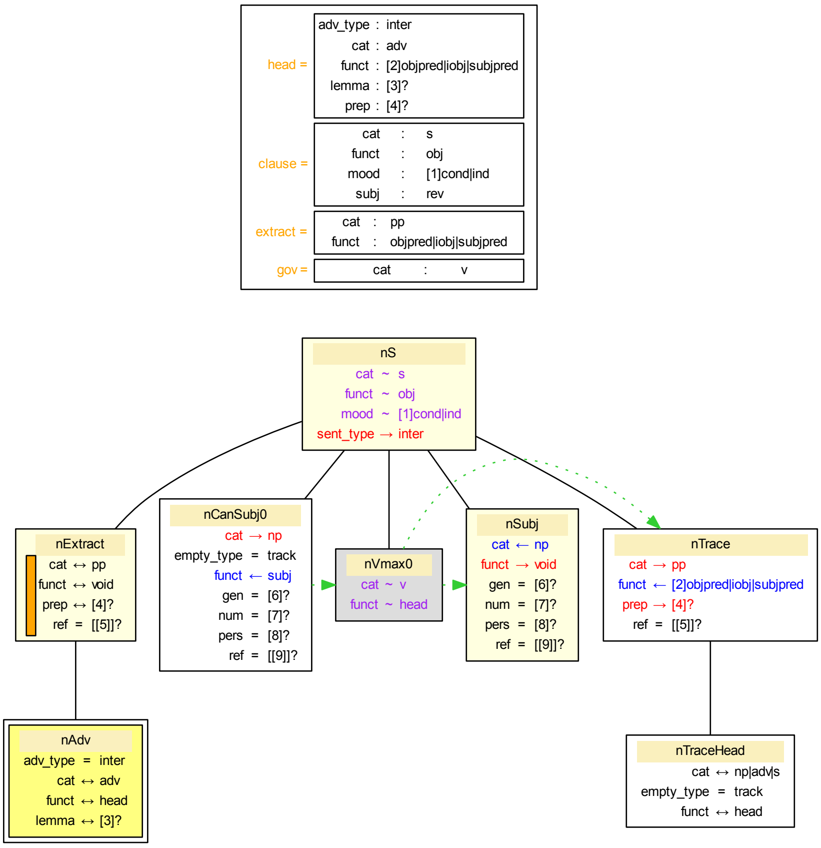

Figure 10.30: An EPTD defined by the ADVINTER_S1 class and used in the parsing of Sentence (10.68).

10.3.3 Interrogative Determiners

The unique interrogative determiner *quel* has multiple uses which are illustrated with the following sentences.

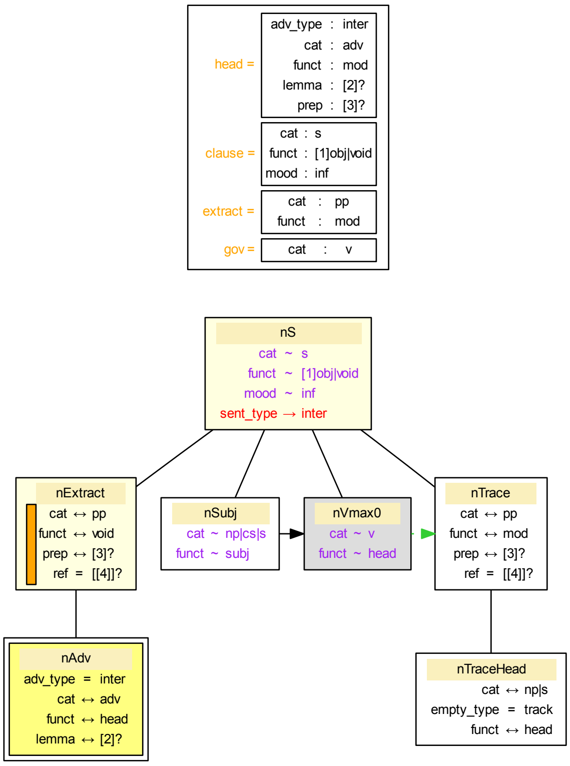

Figure 10.31: An EPTD defined by the ADVINTER_S1 class and used in the parsing of Sentence (10.69).

(10.70) *Jean demande* **quelle** *femme Marie est .*
Jean asks which woman Marie is .
Jean asks which woman Marie is.

(10.71) *Au directeur de* **quelle** *entreprise Marie pense-t-elle que Jean a parlé ?*
To director of which company Marie believes that Jean has told ?
The director of which company Marie believes that Jean told?

(10.72) **Quel** *ami* *pourra* *m'aider* *?*
Which friend can help me ?
Which friend can help me?

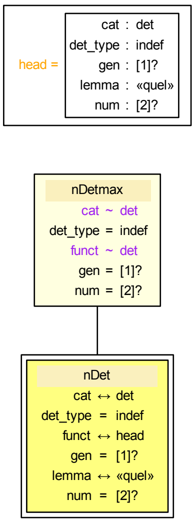

Figure 10.32: The EPTD defined by the InterrogativeDeterminer class.

A basic class InterrogativeDeterminer gives the common skeleton of all EPTDs associated with interrogative determiners. It is the PTD shown in Figure 10.32. Then, the class is specialized in the following sub-classes:

- DETInter_N1dircompl_S2 for extracted constituents that are direct complements. This class is the conjunction of classes InterrogativeDeterminer and InterrogativeWithExtractedComplement. Moreover, it merges nodes *nWh* and *nDetmax* and it adds restrictions on the category and the function of *nTrace*. Figure 10.33 shows an EPTD defined by this class and used in the parsing of Sentence (10.70). In this example, the extracted constituent *quelle femme* is represented by node *nExtract* and the corresponding trace by node *nTrace*. The trace is subject predicative complement of the verb *est*.

- DETInter_N1indircompl_S2 for extracted constituents that are indirect complements. This class is the conjunction of classes InterrogativeDeterminer and InterrogativeWithExtractedComplement. Moreover, it merges nodes *nWh* and *nDetmax* and it adds restrictions on the category and the function of *nTrace*. On the left of Figure 10.34, there is an EPTD defined by this class and used in the parsing of Sentence (10.71). In this example, the extracted constituent *au directeur de quelle entreprise* is represented by node *nExtract* and the corresponding trace by node *nTrace*. The trace is embedded in the object clause *que Jean a parlé* represented by nodes *nS0* and *nS1*, which are merged. The complete sentence is represented by node *nS*.

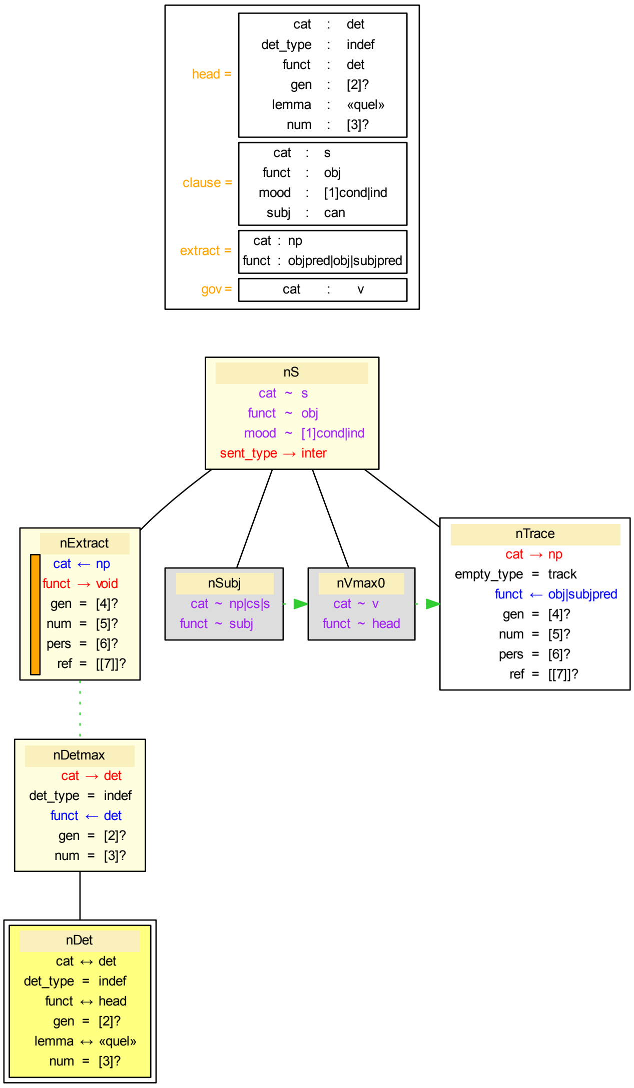

Figure 10.33: EPTD defined by the DETINTER_N1IDIRCOMPL_S2 class and used in the parsing of Sentence (10.70).

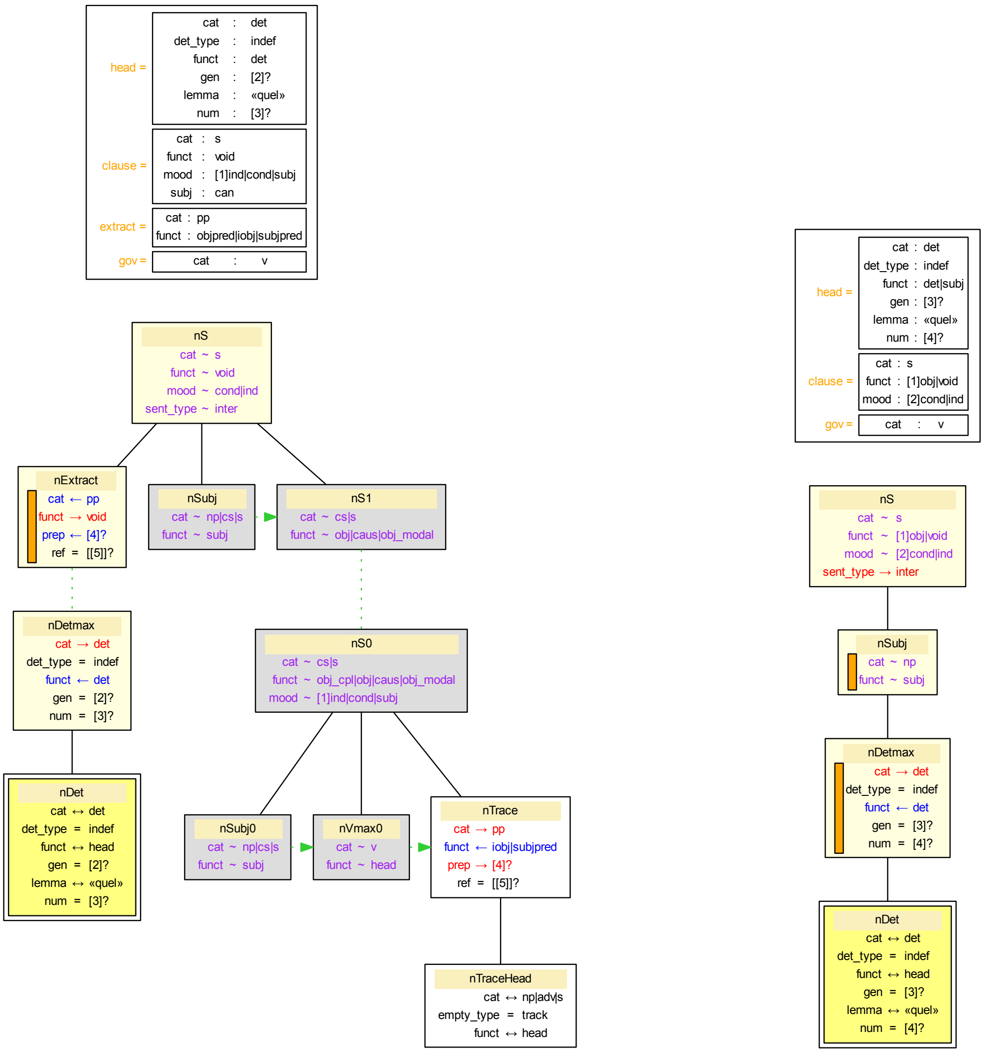

Figure 10.34: EPTDs defined by the DETINTER_N1INDIRCOMPL_S2 and DETINTER_N1SUBJ_S2 classes and used in the parsing of Sentences (10.71) and (10.72).

- DETINTER_N1SUBJ_S2 for constituents with *quel* as determiner that are subjects. The class is the conjunction of classes INTERROGATIVEDETERMINER and NONEXTRACTEDSUBJECT. It defines a unique EPTD shown on the right of Figure 10.34 and used in the parsing of Sentence (10.72). In this example, the subject constituent *quel ami* is represented by node *nSubj*.

10.4 Cleft clauses

Cleft clauses are used to focus on an expression, called the *focus* of the cleft clause. The focus is subject predicative complement of the verb *être* and the subject of the verb is always *ce*. The focus is extracted from a clause, which becomes a subordinated clause depending on the verb *être*. If the extracted constituent is the subject, it is replaced with the relative pronoun *qui*. If the extracted constituent is a complement, the subordinated clause is introduced with the complementizer *que*.

Here are examples of cleft clauses. The focus is in bold and the trace, when it exists, is marked with the symbol □.

(10.73)	*C'*	*est*	**à**	**Eva**	*que*	*Tom*	*croit*	*avoir*	*donné*	*une*	*invitation* □	*.*
	it	is	to	Eva	that	Tom	believes	to have	given	an	invitation	.

It is to Eva that Tom believes to have given an invitation.

(10.74)	*Ce*	*doit*	*être*	**Jean**	*qui*	*vient*	*demain*	*.*
	It	may	be	Jean	who	is coming	tomorrow	.

It may be Jean who is coming tomorrow.

(10.75)	*Est-*	*ce*	**Jean**	*que*	*Marie*	*connaît*	*.*
	is	it	Jean	whom	Marie	knows	.

is it Jean whom Marie knows.

Since FRIGRAM is lexicalized, the information related to cleft clauses must be anchored at words. We have chosen to distribute it between the demonstrative pronoun *ce* and the relative pronoun *qui* or the complementizer *que* introducing the subordinated clause.

A polarized feature `cleft` with the value `true` is used to control the interaction between the two words. The demonstrative *ce* brings a negative feature `cleft` ← `true`, whereas the relative pronoun or complementizer brings the positive feature `cleft` → `true`.

10.4.1 The role of the demonstrative pronoun *ce* in cleft clauses

The CeSubjectClitic class from the proclitic module defines all PTDs associated with *ce* used as a subject clitic pronoun. Among them, there are four PTDs for their use in cleft clauses, according to two parameters: the nature of the grammatical word introducing the subordinated clause and the presence or not of a modal auxiliary modifying the verb *être*.

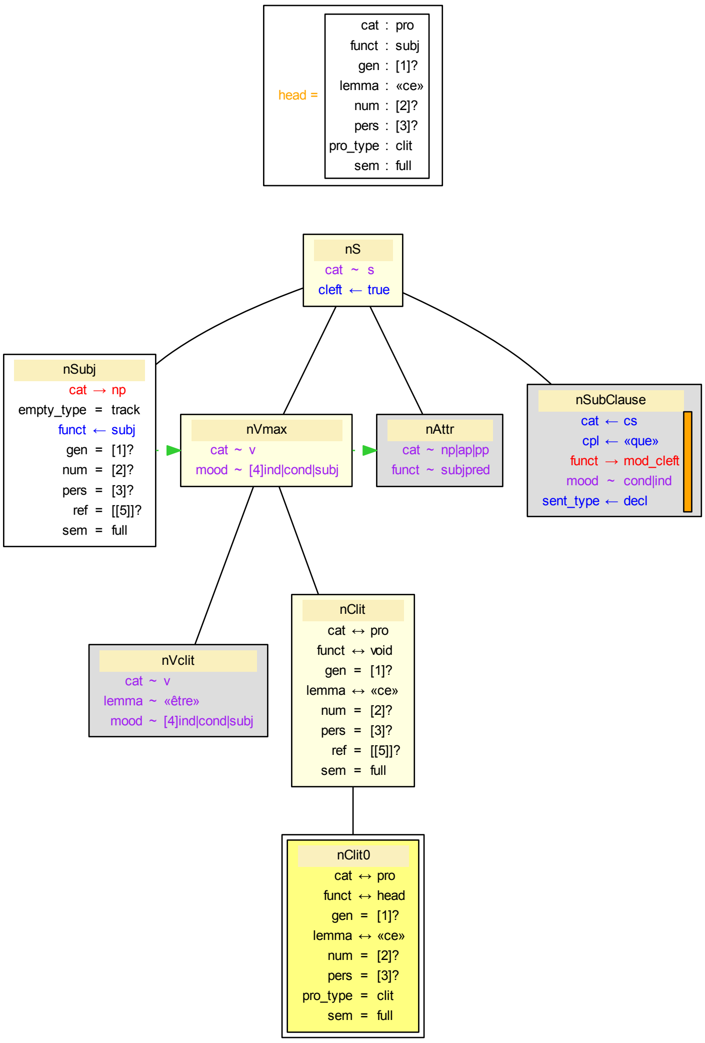

Figure 10.35: PTD defined by the CeSubjectClitic class and used in the parsing of Sentences (10.73) and (10.75).

Figure 10.35 shows the PTD used for *ce* in the parsing of Sentences (10.73) and (10.75). Node *nS* represents the whole cleft clause with its negative feature `cleft` $\leftarrow$ `true`. Node *nClit* represents the clitic pronoun *ce*.

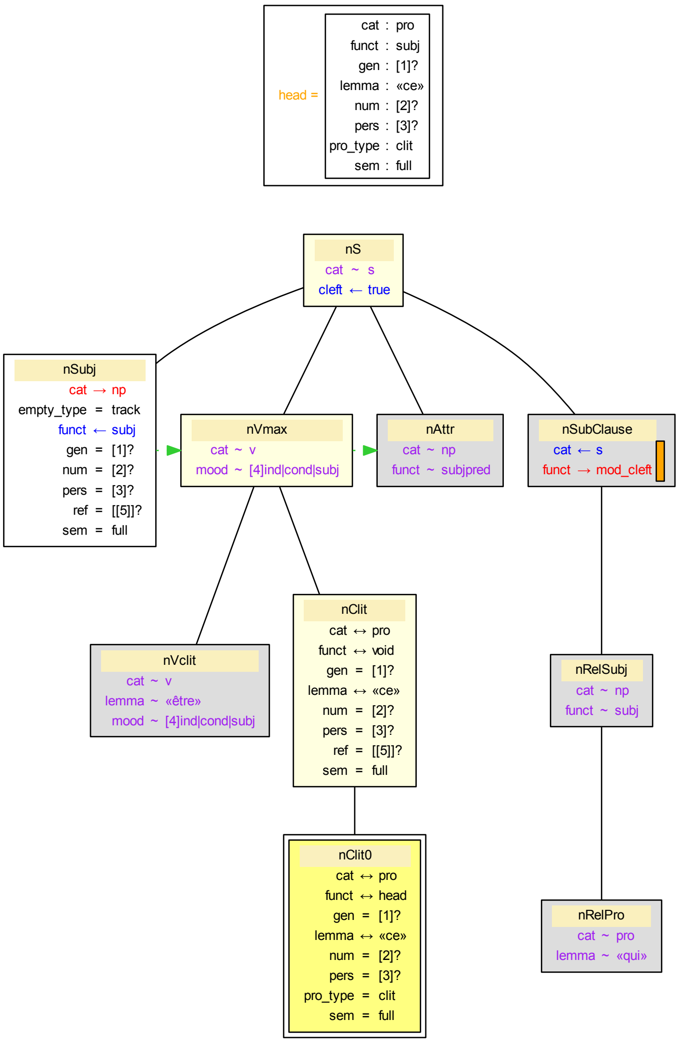

Figure 10.36: PTD defined by the CeSubjectClitic class and used in the parsing of Sentence (10.74).

Since the clitic is not in canonical position, there is an empty trace in this position represented by node *nSubj*. Node *nAttr* represents the focus which is a subject predicative complement of the verb *être* and node *nSubClause* represents the expected subordinated clause introduced with the complementizer *que*[3].

Figure 10.36 shows the PTD used for *ce* in the parsing of Sentence (10.74). The only

[3] In Sentence (10.75), the word *que* can be also interpreted as a relative pronoun, which is not the case in Sentence (10.73).

difference with respect to the previous PTD is *nSubClause*, which represents a relative clause. The subtree rooted at this node expresses the constraint that the subject of the relative clause is the pronoun *qui*.

The CeSubjectClitic class provides four PTDs for the pronoun *ce* used in cleft clauses but in fact, it corresponds to eight EPTDs because the two possibilities of order between the clitic and the verb:

- the order clitic-verb for declarative sentences (Sentences (10.73) and (10.74)) and the PROclit-subj-decl class, which inherits the CeSubjectClitic class expresses it;
- the order verb-clitic for interrogative sentences (Sentence (10.75)) and the PROclit-subj-inter class, which inherits the CeSubjectClitic class expresses it.

10.4.2 The role of the complementizer *que* or the relative pronoun *qui* in cleft clauses

In complement with the EPTD attached at the pronoun *ce*, the EPTD attached at the complementizer *que* or the relative pronoun *qui* brings information related to the cleft clause.

In the extractGramWord module, a basic class CleftClause defines the PTD constituting the skeleton common to the two cases.

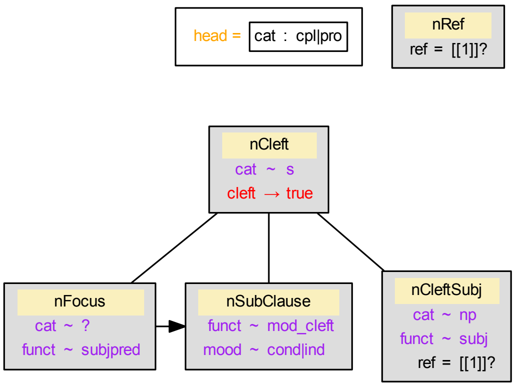

Figure 10.37: PTD defined by the CleftClause class.

Figure 10.37 presents this PTD. Node *nCleft* represents the cleft clause, and three of its sub-constituents, the focus, the subject and the subordinated clause, are respectively represented by nodes *nFocus*, *nCleftSubj* and *nSubClause*. Node *nRef* co-refers with *nCleftSubj* and in case of extraction, it represents the trace of the extracted constituent; otherwise, it represents the relative pronoun *qui*.

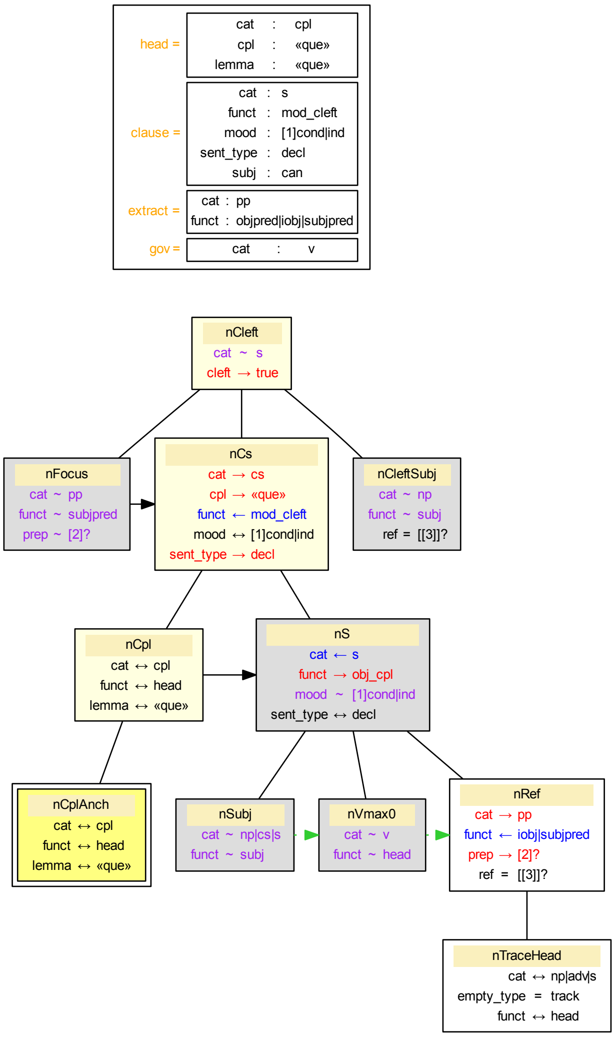

Figure 10.38: EPTD defined by the CPL_S1CLEFT class and used in the parsing of Sentence (10.73).

The CleftClause is specialized in two sub-classes:

- CleftClauseExtractedComplement, when the subordinated clause is introduced by the complementizer *que*. The class inherits a disjunction of the following classes: ExtractedRequiredVerbComplement, ExtractedAdjunctVerbCom-

PLEMENT and EXTRACTEDADJECTIVECOMPLEMENT. Each of them expresses a particular function of the extracted constituent. Moreover, the class merges nodes *nFocus* and *nExtract* as well as nodes *nRef* and *nTrace*. It also makes node *nS* daughter of node *nSubClause*.

- CLEFTCLAUSESUBJECT, when the subordinated clause is introduced by the relative pronoun *qui*. The class is the conjunction of NONEXTRACTEDSUBJECT and CLEFTCLAUSE. in addition, it merges nodes *nRef* and *nSubj*, as well as nodes *nS* and *nSubClause*.

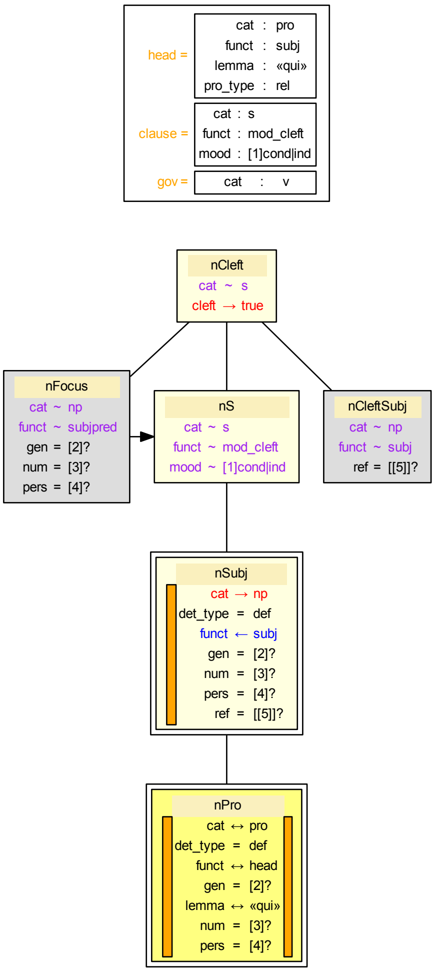

Figure 10.39: EPTD defined by the PROREL-SUBJ_S1CLEFT class and used in the parsing of Sentence (10.74).

In the COMPLEMENTIZER module, the CPL_S1cleft class defines the 18 EPTDs attached at the complementizer *que* used in a cleft clause. It inherits the Complementizer and CleftClauseExtractedComplement classes. Among the 18 EPTDs, that one used in the parsing of Sentence (10.73) is shown in Figure 10.38.

In the RELATIVE module, the PROrel-subj_S1cleft class defines a unique EPTD attached at the relative pronoun *qui* used in a cleft clause. It inherits the CleftClauseSubject class and the EPTD that it defines is shown in Figure 10.39.

10.4.3 The expression *est-ce que*

In FRIGRAM, *est-ce que* is considered as a complex expression, and in some cases, it is used in interrogative cleft clauses. Here are various uses of this expression. When there is extraction, the trace of the extracted constituent is marked with □ and in case of multiple extraction, the different traces are indexed.

(10.76) *Est -ce que Jean vient ?*
Is it that Jean is coming ?

Is Jean coming?

(10.77) *Quand est -ce* $\square_1$ *que Jean vient* $\square_2$ *?*
when is it that Jean is coming ?

When is Jean coming?

(10.78) *qui est -ce* □ *qui vient ?*
Who is it who is coming ?

who is coming?

(10.79) *qui est -ce* $\square_1$ *que Jean voit* $\square_2$ *?*
Whom is it that Jean sees ?

whom does Jean see?

Sentence (10.76) can be interpreted as the interrogative counterpart of the declarative sentence *c'est que Jean vient*, where *que Jean vient* is a subject predicative complement of *est*.

The three other sentences can be interpreted as interrogative cleft clauses. For Examples (10.77) and (10.79), there are two extractions related to the cleft clause and to the interrogation. For Example (10.78), there is only one extraction because in the cleft clause, the relative pronoun *qui* replaces the extracted constituent in situ.

10.5 Dislocation

The cleft construction is a way of focusing on a particular element of sentence. Dislocation is another way of doing the same by separating the element at the beginning or at the end of the sentence and by resuming it with a clitic pronoun. Here are examples of dislocations (the dislocated element is between square brackets and the resumption pronoun is in bold).

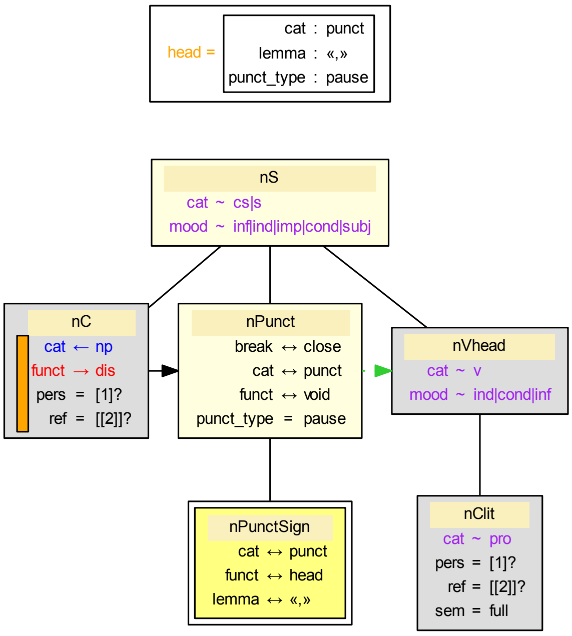

Figure 10.40: EPTD defined by the PUNCTCLOSE_NP1LEFT class and used in the parsing of Sentence (10.80).

(10.80) *[La mer]* *,* *Jean* **en** *rêve* *.*
The sea , Jean of it dreams .
The sea, Jean dreams of it.

(10.81) **Elle** *est* *belle* *,* *[la* *mer]* *.*
It is beautiful , the sea .
It is beautiful, the sea.

(10.82) **C'** *est* *impossible* *[que* *je* *vienne]* *.*
It is impossible that I come .
It is impossible that I come.

(10.83) **C'** *est* *impossible* *[de* *venir]* *.*
It is impossible to come .
It is impossible to come.

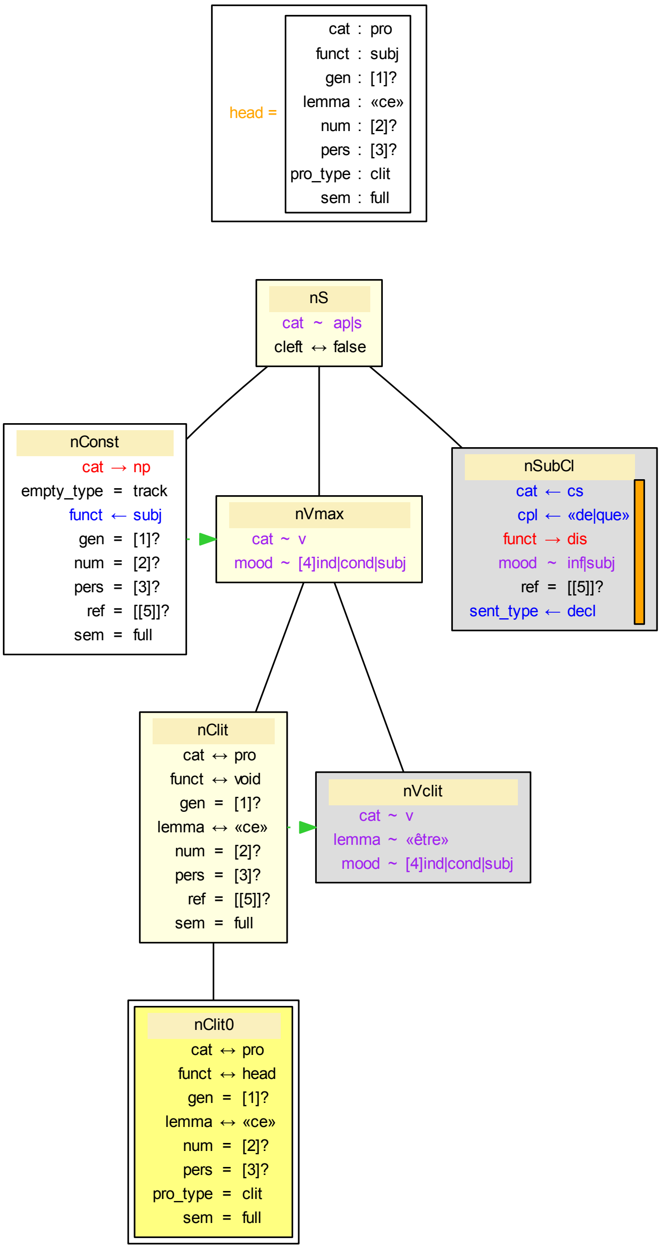

Figure 10.41: EPTD defined by the PROclit-subj-decl class and used in the parsing of Sentences (10.82) and (10.83).

Sentences (10.80) and (10.81) show a first case of dislocation, where the dislocated element is separated with a comma. In FRIGRAM, the construction is attached at the comma. The comma anchors the EPTD generated by the PUNCTclose_NP1left class, shown on Figure 10.40 and used in the parsing of Sentence (10.80).

Sentences (10.82) and (10.83) show a second case of dislocation, where the dislocated element is a infinitive clause introduced with *de* or a finite clause introduced with *que*. The resumption pronoun is necessarily *ce*. The EPTD used to parse Sentences (10.82) and (10.83) is generated by the PROclit-subj-decl class and shown on Figure 10.41. It is anchored by the pronoun *ce*. Node *nConst* represents the empty trace of the clitic *c'*, whereas node *nSubCl* represents the dislocated clause.

Chapter 11

Coordination and Punctuation

Punctuation signs in FRIGRAM are considered as words and, since FRIGRAM is lexicalised, all grammatical information related to punctuation must be attached at particular punctuation signs.

Coordination and punctuation have the following property in common that their behavior is very particular with respect to other grammatical phenomena. The number of syntactic constructions that they give rise to is very high. Maybe it is better to take them into account outside the grammar in the parsing process.

To integrate the two phenomena inside our grammar FRIGRAM with a relatively large coverage would entail an explosion of the size of the grammar. Therefore, we have decided to integrate them in a very limited way by restricting the coverage to the simplest cases compatible with the formalism of IG.

11.1 Coordination

Coordination is a polymorphic phenomenon in the sense that various types of expressions can be coordinated. Here are a few examples of coordination that FRIGRAM is able to process, with the EPTDs of the Coordination module. The conjunction of coordination is in bold and the two conjuncts are between square brackets.

(11.1)	*[Marie]*	**et**	*[Pierre]*	*viennent*	*demain*	*.*
	Marie	and	Pierre	are coming	tomorrow	.

Marie and Pierre are coming tomorrow.

(11.2)	*Marie*	*est*	*[intelligente]*	**et**	*[décidée]*	*.*
	Marie	is	clever	and	resolute	.

Marie is clever and resolute.

(11.3)	*Je*	*vais*	*à*	*Paris*	*pour*	*[travailler]*	**et**	*[rejoindre]*	*Marie*	*.*
	I	am going	to	Paris	to	work	and	to join	Marie	.

I am going to Paris to work and to join Marie.

(11.4) *[Marie vient demain]* **et** *[Pierre part aujourd'hui]* .
Marie is coming tomorrow and Pierre is leaving today .
Marie is coming tomorrow and Pierre is leaving today.

(11.5) *Jean pense [que Marie vient demain]* **et** *[que Pierre part aujourd'hui]* .
Jean believes that Marie is coming tomorrow and that Pierre is leaving today .
Jean believes that Marie is coming tomorrow and that Pierre is leaving today.

(11.6) *Il [vient aujourd'hui]* **et** *[rencontre Marie]* .
He is coming today and is meeting Marie .
He is coming today and he is meeting Marie.

(11.7) *Paul a appris [l'espagnol]* **,** *[l'italien]* **,** *[le portugais]* .
Paul has learnt Spanish , Italian , Portugese .
Paul has learnt Spanish, Italian, Portugese.

In FRIGRAM, we have chosen to take the conjunction of coordination as the head of the coordination and to consider that the coordination distributes its syntactic function to its conjuncts via the feature `funct`. In the model, if the node representing the coordination has the feature `funct : X`, the nodes representing the two conjuncts have the same feature.

Enumeration, illustrated with Example (11.7), is considered as a particular case of coordination, with comma playing the role of a coordination conjunction.

The basic class ArgumentCoord models these properties. In the interface, the feature **`head`** gives the properties of the conjunction of coordination, whereas the feature `conj1` gives the properties of the conjuncts.

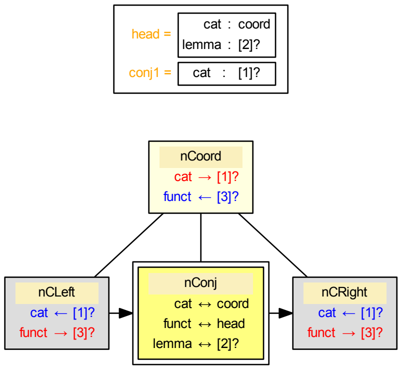

Figure 11.1: The PTD defined by the ArgumentCoord class

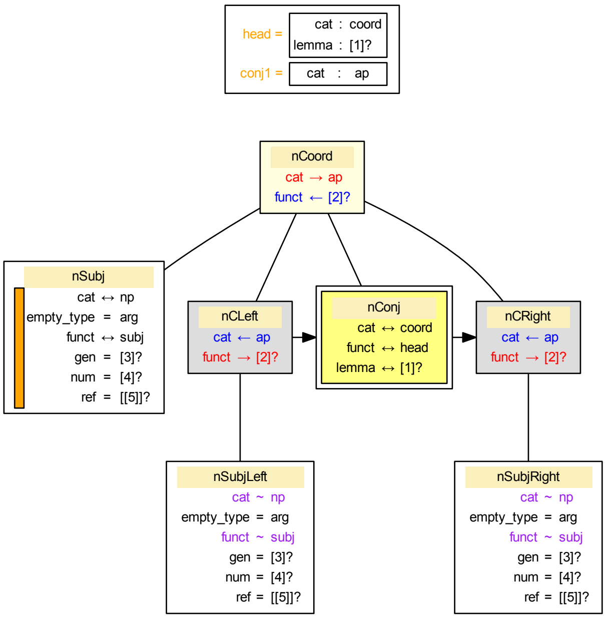

Figure 11.2: The EPTD defined by the AP0_CONJ_AP1 class

Then, the class is specialized in different sub-classes according to the category of the conjuncts:

- NP0_CONJ_NP1 when the conjuncts are noun phrases (Example (11.1)); the only agreement rule that is modeled in this case is that the the coordination is in the plural when the conjunction is *et*.

- AP0_CONJ_AP1 when the conjuncts are adjectival phrases with a function of predicate(Example (11.2)). Figure 11.2 shows the EPTD defined by the class. The two conjuncts share the same subject; hence, nodes *nSubjLeft* and *nSubjRight* corefer to the same entity, which is expressed with the feature `ref`. In the interaction of the EPTD with other EPTDs in the parsing process, the common subject plays a role; hence it is necessary to have a copy of it in the canonical position. Node *nSubj* represents this copy.

- S0inf_CONJ_S1inf, when the conjuncts are infinitives (Example (11.3)). The class defines an EPTD, which is similar to that of Figure 11.2, because the conjuncts also share a common subject.

- S0fin-decl_CONJ_S1fin-decl when the conjuncts are finite declarative sentences (Example (11.4)). The two nodes representing the conjuncts are provided with a saturated feature sent_type ↔ decl, whereas the node representing the coordination has the virtual feature sent_type ∼ decl. Regarding the mood, we impose that the feature mood be virtual for the conjuncts and saturated for the coordination; moreover, the only value possible is one of the two values cond or ind.

- S0FIN-IMP-INTER_CONJ_S1FIN-IMP-INTER when the conjuncts are finite imperative or interrogative sentences. The two nodes representing the conjuncts are provided with a negative feature sent_type ← imper|inter, whereas the node representing the coordination has the positive feature sent_type → imper|inter. Regarding the mood, we impose that the feature mood be virtual for the conjuncts and saturated for the coordination; moreover, the only value possible is one of the three values bond, imper or ind.

- CS0_CONJ_CS1, when the conjuncts are complemented clauses (Example (11.5)). Figure 11.3 shows the EPTD defined by the class. It imposes that the two complementizers be the same, which is a stronger constraint than in the reality.

- VP0_CONJ_VP1, when the conjuncts are verb phrases (Example (11.6)). Figure 11.4 shows the EPTD defined by the class. Contrary to the previous EPTDs, this one introduces no symmetry between the two conjuncts. Since in FRIGRAM there is no notion of verb phrase, we coordinate two sentences, the second one where the subject is elided and represented with an empty node *nSubjR*.

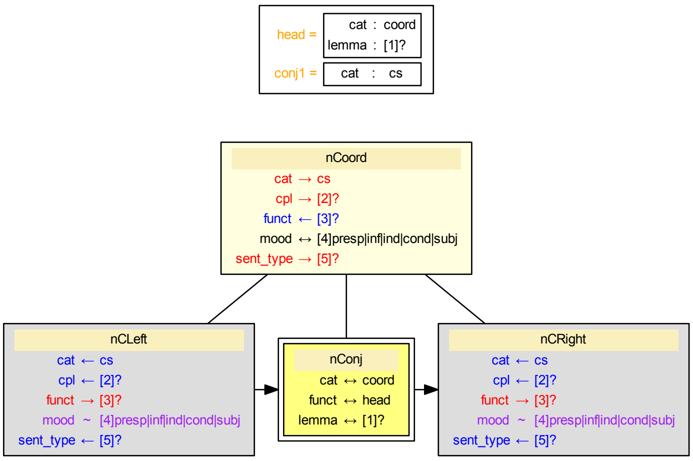

Figure 11.3: The EPTD defined by the CS0_CONJ_CS1 class

In FRIGRAM, it is not possible to model the coordination of all types of constituents having a function of argument. If a type of constituent is represented in FRIGRAM not with a single node but with a tree, the EPTD associated with the conjunction of coordination must introduce a copy of this tree, which is not motivated linguistically and complicates the representation.

Another problem comes from the coordination of modifiers because in the EPTD associated with a modifier a root represents the constituent to be modified, and there is no way of transferring the root from the conjuncts to the coordination.

Sometimes, conjunctions of coordination are used without a first conjunct, as the following example shows it.

(11.8) *Et si on allait se balader ?*
And if we will go to stroll ?
And if we will go to stroll?

The CONJ_S1FIN class defines the EPTD corresponding to this particular use of conjunctions of coordination.

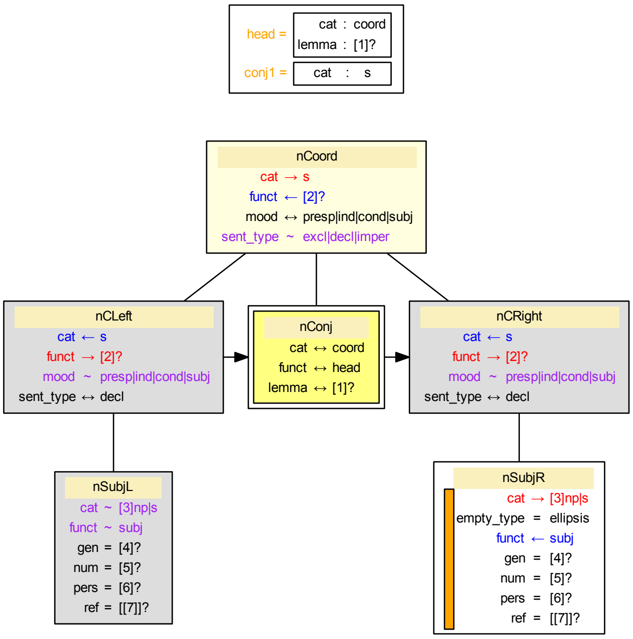

Figure 11.4: The EPTD defined by the VP0_CONJ_VP1 class

11.2 Punctuation

Punctuation signs are considered as words and FRIGRAM associates EPTDs to them in the particular module PUNCTUATION. The PUNCTUATION class defines the common skeleton of all these EPTDs.

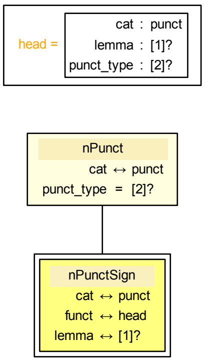

Figure 11.5: The EPTD defined by the PUNCTUATION class

In the interface of the EPTD, the feature `head.punct_type` gives the type of the punctuation sign: `stop` for signs ending a sentence, `pause` for signs marking a pause inside a sentence. Then, the PUNCTUATION class is specialized in four kinds of classes:

- classes for signs ending a sentence,
- classes for commas marking the end of a detachment at the beginning of a sentence,
- classes for commas introducing or closing an apposition or an insertion,
- classes for specific marks at the end constituents, when these constituents are parsed in a isolated way outside a sentence.

As we said previously, commas used in enumerations are considered as conjunctions of coordination.

11.2.1 Signs ending a sentence

The STOP class is a specialization of the PUNCTUATION class for all signs marking the end of a sentence

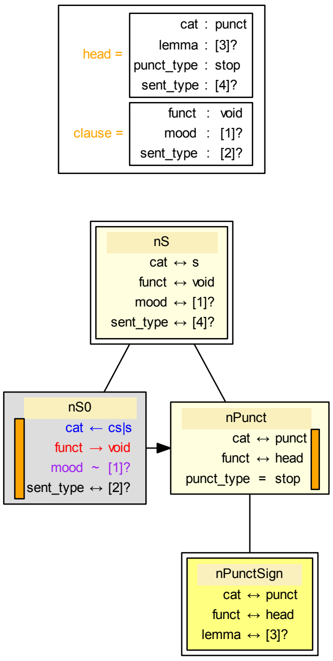

Figure 11.6: The EPTD defined by the Stop class

Figure 11.6 presents the EPTD defined by the Stop class. A sentence ended with a punctuation sign is considered as a new constituent represented by node *nS*, with respect to the bare sentence represented by node *nS0*. The new level is required by the occasional need of expressing that a constituent is a the end of a sentence and it is not possible if a punctuation sign occupies the position at the end of the sentence.

There is a co-indexation between the feature `head.sent_type` of the interface and the feature `sent_type` of node *nS*. There is the same between the feature `clause.sent_type` of the interface and the feature `sent_type` of node *nS0*. A difference may occur between the type of *nS* and the type of *nS0*, hence two separate co-indexations. For instance, in Example (11.10) below, the bare sentence *Appeler demain* is declarative and the complete sentence *Appeler demain !*

The Stop class is then specialized according to the type of the punctuation sign. Here are different examples which illustrate various cases.

(11.9) *Jeanne dort .*
Jeanne is sleeping .
Jeanne is sleeping.

(11.10) *Appeler Jeanne demain !*
To call Jeanne tomorrow !

To call Jeanne tomorrow!

(11.11) *Jeanne dort -elle ?*
Jeanne is sleeping she ?

Is Jeanne sleeping?

(11.12) *Dormez bien !*
Sleep well !

Sleep well!

(11.13) *Qu' elle dort bien !*
she is sleeping well !

She is sleeping so well!

(11.14) *Que Jeanne dorme tranquille !*
that Jeanne sleep quietly !

Let Jeanne sleep quietly!

(11.15) *Jeanne entre : tout le monde se tait .*
Jeanne is entering : all people shut up .

Jeanne is entering: all people shut up.

The difference between the sub-classes of Stop come from the differences in which the polarized feature `sent_type` becomes saturated.

- For the PUNCTstop_S1decl class (Sentences (11.9) and (11.10)), the bare sentence is declarative and its definitive type is given by the single punctuation sign, so that the class brings a saturated feature `sent_type` ↔ `decl` to *nS0*.

- For the PUNCTstop_S1nondecl class, a word in the sentence brings its type in the form of a positive feature `sent_type` to *nS0*: it may be the subject clitic which comes after the verb (Sentence (11.11)), or a verb in the imperative mood (Sentence (11.12)) or the conjunction *que* (Sentence (11.13)). The EPTD of the punctuation sign brings the negative dual feature `sent_type`, to *nS0*, which will neutralize the first one.

- For the PUNCTstop_CS1inter-imper, *nS0* is a complemented sentence. On its left, Figure 11.7 shows the EPTD defined by the class and used in the parsing of Sentence (11.14).

Sentence (11.15) illustrates a use of the colon to separate two sentences. The EPTD anchored by the colon is defined by the PUNCTcolon_S1_S2 class and shown on the right of Figure 11.7.

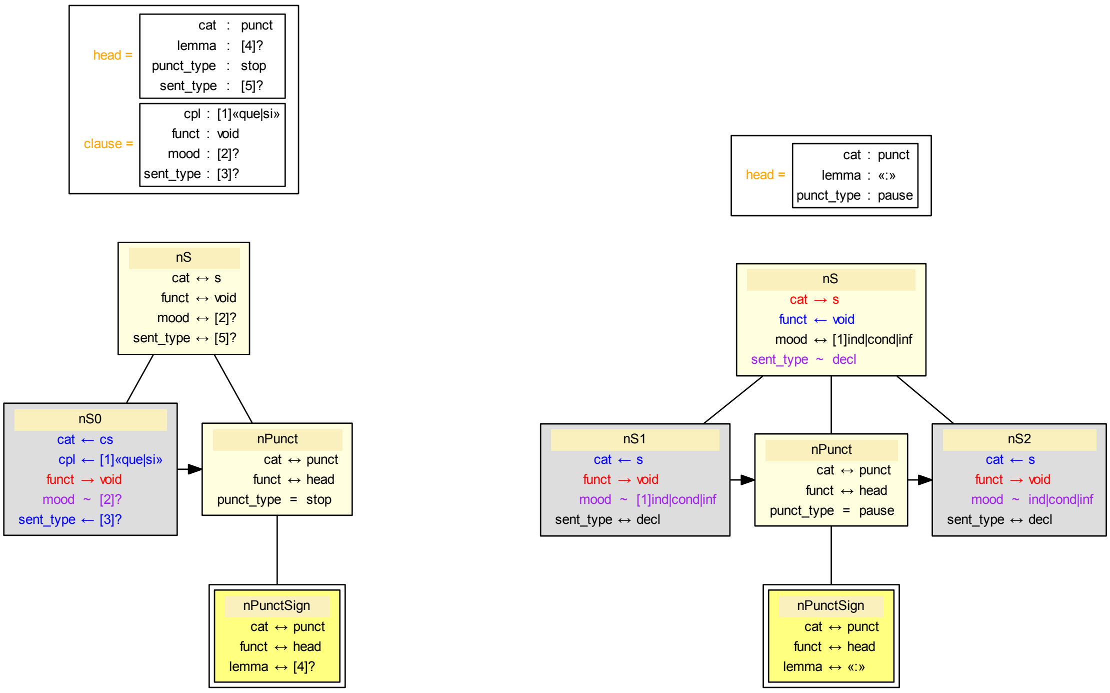

Figure 11.7: The EPTDs defined by the PUNCTSTOP_CS1INTER-IMPER and PUNCT-COLON_S1_S2 classes

11.2.2 Commas marking the end of a detachment at the beginning of a sentence

Here are sentences illustrating different uses of commas marking the end of a detachment at the beginning of a sentence.

(11.16) *Aujourd'hui , Jeanne travaille jusqu'au soir .*
Today , Jeanne is working until tonight .
Today, Jeanne is working until tonight.

(11.17) *Jeanne étant arrivée , Marie peut partir .*
Jeanne having arrived , Marie may leave .
Jeanne having arrived, Marie may leave.

(11.18) *Venant de l' entreprise , Pierre veut voir Jeanne .*
coming from the company , Pierre wants to see Jeanne .
coming from the company, Pierre wants to see Jeanne.

(11.19) *Jeanne , elle vient demain .*
Jeanne , she is coming tomorrow .
Jeanne, she is coming tomorrow.

(11.20) *Jeanne , viens !*
Jeanne , come !
Jeanne come!

A basic class, ClosingDetachmentComma, defines the common skeleton of all EPTDs used for the comma in the parsing of the examples above. This skeleton is shown in Figure 11.8. The class inherits the Puncutation class.

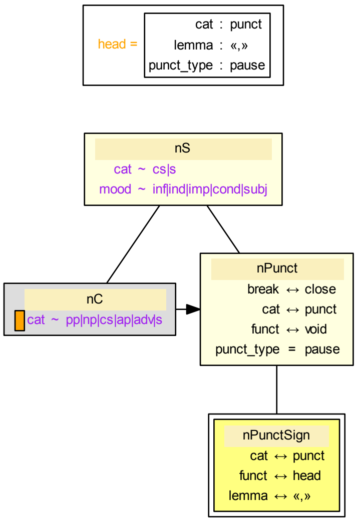

Figure 11.8: The EPTDs defined by the ClosingDetachmentComma class

Node *nS* represents the whole sentence and node *nC* represents the detached constituent.

The ClosingDetachmentComma is specialized in the following sub-classes:

- PUNCTclose_C1left when *nC* is a prepositional phrase or complemented clause that modify the main clause (Example (11.16)). The class adds two features to node *nC*: `cat` ∼ `cs|pp` and `funct` ∼ `mod`.

- PUNCTclose_S1left when *nC* is a clause in a participle mood that modify the main clause (Example (11.17)). The class adds three features to node *nC*: `cat` ← `s`, `mood` ∼ `pastp|presp` and `funct` → `mod`.

- PUNCTclose_AP1left when *nC* is a clause in an adjectival phrase that modify the main clause and refers to its subject (Example (11.18)). Figure 11.9 shows the EPTD defined by the class. Node *nSubj2* represents the empty subject of the detached adjectival phrase and it co-refers with node *nSubj1*, which represents the subject of the main clause.

- PUNCTclose_NP1left when *nC* is a noun phrase that is a dislocated complement of the main verb and it is repeated by a clitic. In Example (11.19), the dislocated noun phrase *Jeanne* is repeated by the subject clitic *elle*. Figure 11.10 shows the EPTD defined by the class. Node *nClit* represents a clitic that is a repetition of the detached noun phrase *nC* and it co-refers with this one.

- PUNCTclose_NP1apost when *nC* is a noun phrase with the function of apostrophe (Example (11.20)). The class adds the feature `sent_type` ∼ `excl|imper` to node *nS* and two features to node *nC*: `cat` ← `np` and `funct` → `apps`.

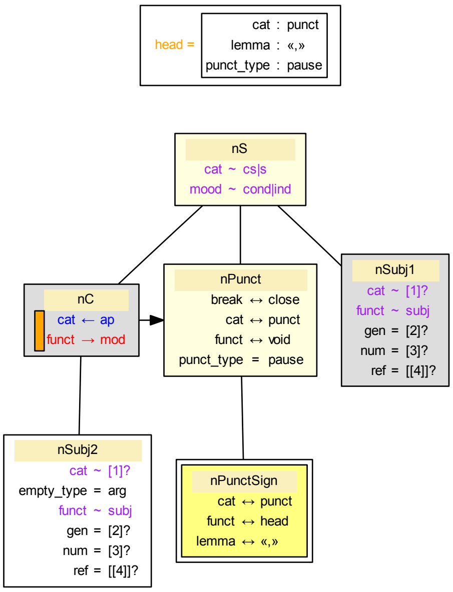

Figure 11.9: The EPTDs defined by the PUNCTclose_AP1left class

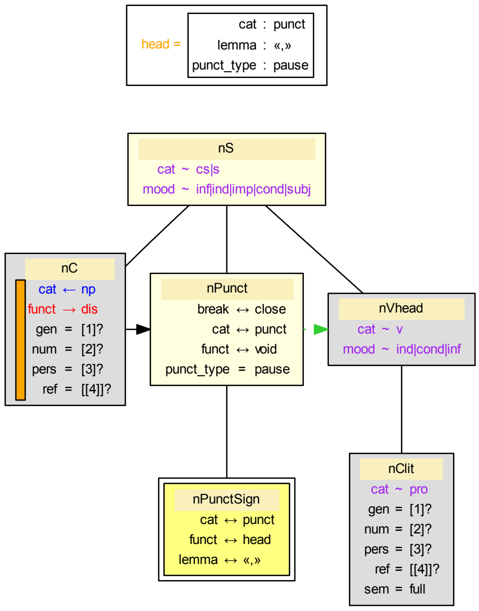

Figure 11.10: The EPTDs defined by the PUNCTclose_NP1left class

11.2.3 Commas introducing or closing an apposition or an insertion

Here are examples illustrating different cases of use for commas introducing or closing an apposition or an insertion.

(11.21) *J' ai rencontré Jeanne , la directrice de l' entreprise , en*
I have met Jeanne , the director of the company , by
venant .
coming .
I have met Jeanne, the director of the company, by coming.

(11.22) *Marie , très gentille , cède sa place à Jeanne .*
Marie , very nice , gives up her place to Jeanne .
Marie, very nice, gives her place up to Jeanne.

(11.23) *J' ai demandé à Marie , qui travaille avec mon père , si*
I have asked to Marie , who works with my father , if
elle venait demain .
she is coming tomorrow .
I have asked to Marie, who works with my father, if she is coming tomorrow.

(11.24) *Jeanne , dans l' entreprise , travaille beaucoup .*
Jeanne , in the company , works a lot .
Jeanne , in the company , works a lot.

(11.25) *Marie , Jeanne le dit , est une femme intelligente .*
Marie , Jeanne it says , is a woman clever .
Marie, Jeanne says it, is a clever woman.

The OpeningInsertionComma class defines the common skeleton of all EPTDs anchored by commas, opening inserted expressions. Figure 11.11 shows this skeleton. Node *nC0* represents the constituent in which the sub-constituent represented by *nC* and introduced by the comma *nPunct*.

If an inserted expression is always limited on the left with a comma, this is not the case on the right because the limit can be the end of the sentence. With IG, it is not possible to express that a constituent is limited on the right by the end of the sentence because the constituent may be embedded more or less deeply inside the sentence as the following example illustrates it.

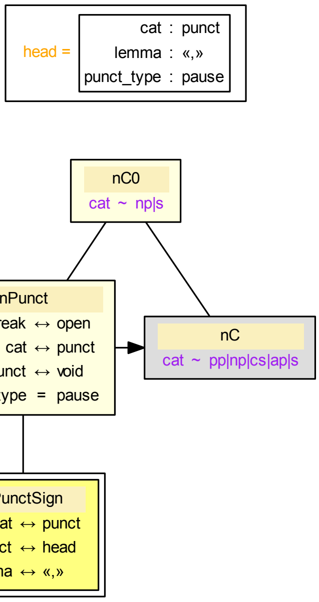

Figure 11.11: The EPTD defined by the OpeningInsertionComma class

(11.26) *Je pense que Marie connaît la maison de Pierre , un ami .*
I believe that Marie knows the house of Pierre , a friend .
I believe that Marie knows the house of Pierre, a friend.

The right limit of the apposition *un ami* is the full stop but it is not at the same level as the apposition in the parse tree of the sentence and in IG, we cannot express a precedence relation between two constituents that are not at the same level.

According to the nature to the insertion, we distinguish insertions that always end with a comma from insertions that may also end with a full stop, but the distinction is not completely right. The OpeningInsertionComma class is specialized in the following sub-classes to take into account the differences illustrated with the sentences above.

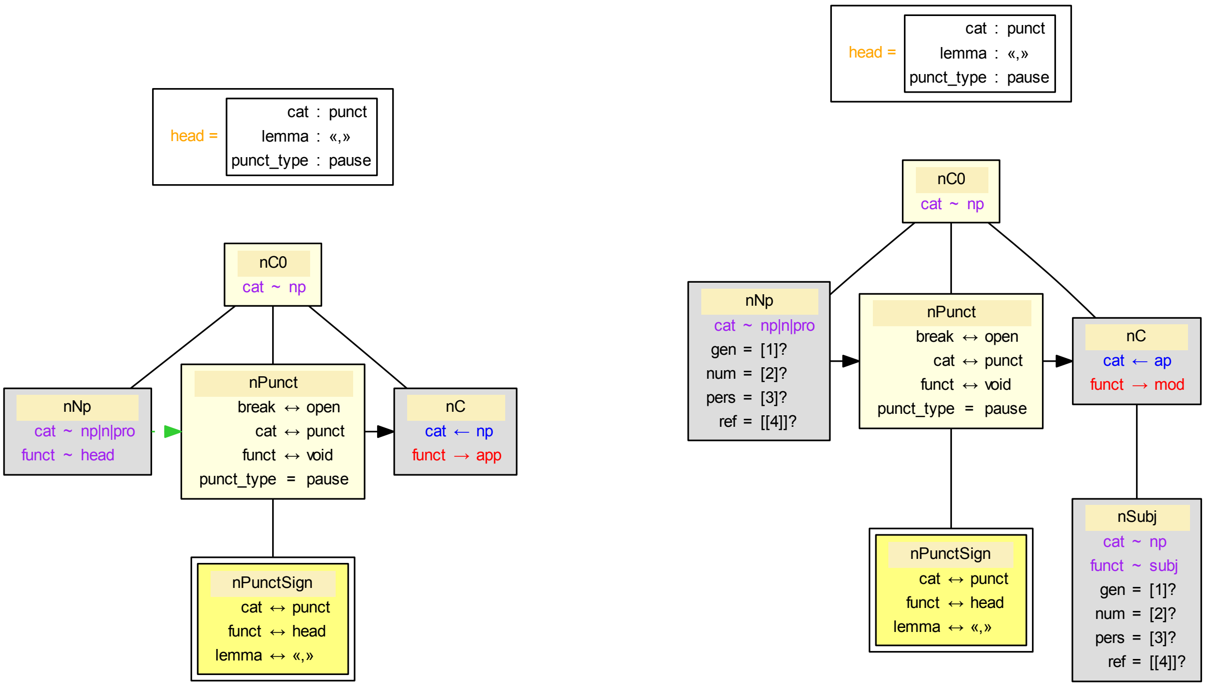

Figure 11.12: The EPTDs defined by the PUNCTOpen_NP1 and PUNCTOpen_AP1 classes

- PUNCTOpen_NP1, when the inserted constituent *nC* is a noun phrase in apposition with respect to an antecedent. Figure 11.12, on its left, shows the EPTD defined by the class and used in the parsing of Sentence (11.21). Node *nNp* represents the antecedent of the noun phrase in apposition, *Jeanne* in the example.

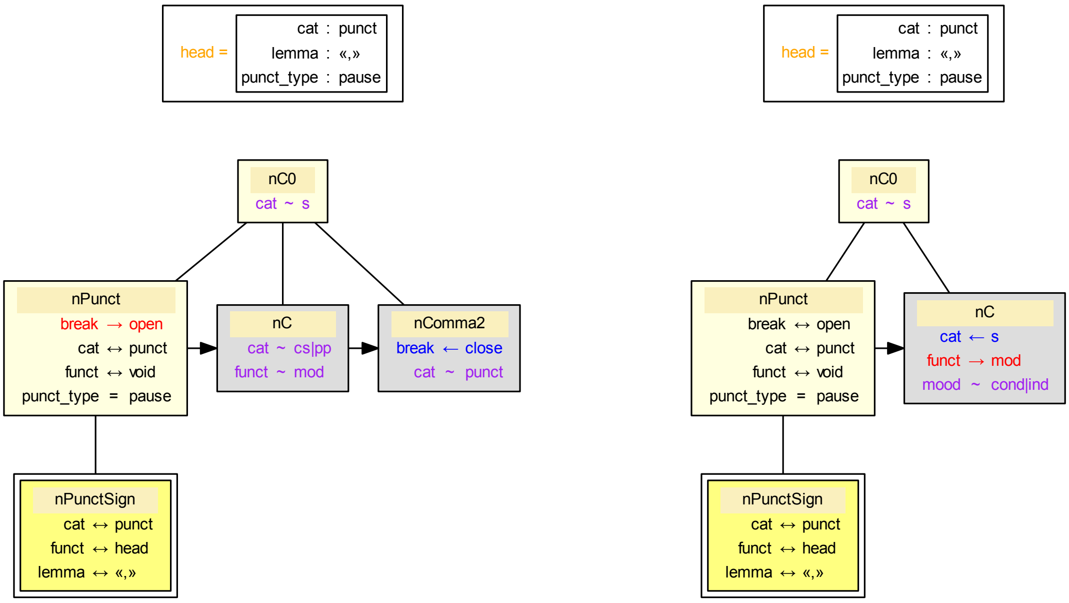

Figure 11.13: The EPTDs defined by the PUNCTOPEN_C1MOD and PUNCTOPEN_S1 classes

- PUNCTOPEN_AP1, when the inserted constituent *nC* is a predicate in the form of an adjectival phrase or a common noun referring to a noun phrase preceding the comma. Figure 11.12, on its right, shows the EPTD defined by the class and used in the parsing of Sentence (11.22). The difference with respect to the previous class is that the inserted predicate has a subject represented by node *nSubj*, which co-refers with the antecedent, represented by node *nNp*.

- PUNCTOPEN_S1REL, when the inserted constituent *nC* is an appositive relative clause (Example (11.23)).

- PUNCTOPEN_C1MOD, when the inserted constituent *nC* is a prepositional phrase or a complemented clause with the function of modifier with respect to *nC0*, which is a sentence (Example (11.24)). Figure 11.13 on its left shows the EPTD defined by the class.

  Contrary to the previous classes, we consider that the inserted prepositional phrase is always surrounded by two commas and the interaction between the two commas is modeled with a polarized feature `break`. The opening comma anchoring the EPTD of Figure 11.24 is represented by node *nPunct*, which carries a positive feature `break` → `open`. The closing comma is represented by node *nComma2* and the EPTD brings a negative feature `break` ← `close` to this node. That is the contrary for the EPTD of the closing comma as we will see later.

- PUNCTopen_S1, when the inserted constituent nC is a parenthetical clause (Example (11.25)). Contrary to the previous case, we do not consider that a parenthetical clause is always closed with a comma: sometimes, it may be put at the end of a sentence. Therefore, nothing is said in the EPTD about what is put after nC.

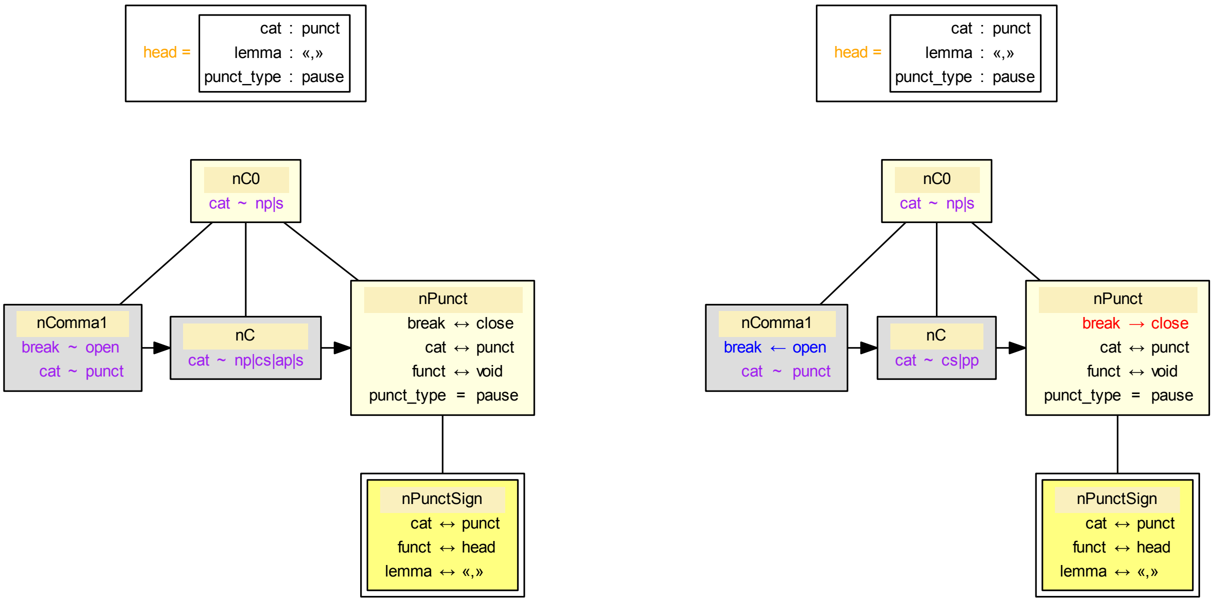

Figure 11.14: The EPTDs defined by the PUNCTclose_C1insert and PUNCTclose_C1mod-insert classes

After commas opening inserted expressions, there are commas closing the same expressions. Since there are two cases, according to fact that the closing comma is required or not by the opening comma, there are two classes:

- PUNCTclose_C1insert for commas closing inserted expressions that are not required by the opening comma (appositions, parenthetical clauses). The inserted expression may be also closed by the end of the sentence. The class defines the EPTD shown on the left of Figure 11.14.

- PUNCTclose_C1mod-insert for commas closing inserted expressions that are required by the opening comma (circumstantial complements). The class defines the

EPTD shown on the right of Figure 11.14.

11.2.4 Signs ending constituents

Sometimes, it is useful to parse single constituents that are neither sentences and nor part of sentences. Since in IG, the models of PTDs must be saturated trees, we add artificial signs that are able to complete PTDs representing the syntax of constituents in order to build saturated trees. Of course, there is a particular sign for each particular constituent.

A class ConstituentStop gives the common skeleton to all concerned PTDs. Figure 11.15 shows this skeleton. The concerned constituent is represented by node *nC*.

The class is specialized in three classes, according to the nature of the constituent: PUNCTstop_C1 for noun and adjectival phrases, PUNCTstop_PP1 for prepositional phrases, PUNCTstop_N1 for common nouns.

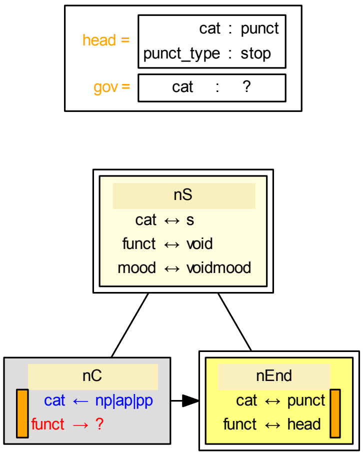

Figure 11.15: The EPTD defined by the ConstituentStop class

Bibliography

[CDG$^+$13] Benoît Crabbé, Denys Duchier, Claire Gardent, Joseph Le Roux, and Yannick Parmentier. XMG : eXtensible MetaGrammar. *Computational Linguistics*, 39(3):1–66, 2013.

[GP09] B. Guillaume and G. Perrier. Interaction Grammars. *Research on Language and Computation*, 7:171–208, 2009.

RESEARCH CENTRE
NANCY – GRAND EST

615 rue du Jardin Botanique
CS20101
54603 Villers-lès-Nancy Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399