

Inférences aspecto-temporelles analysées avec la Logique Combinatoire

Hee-Jin RO¹

(1) Laboratoire LaLIC (Langues, logiques, Informatique et Cognition), Université Paris-Sorbonne, Maison de la Recherche, 28 rue Serpente, 75006 Paris
heejinro@yahoo.fr

Résumé Ce travail s'inscrit dans une recherche centrée sur une approche de l'Intelligence Artificielle (IA) et de la linguistique computationnelle. Il permet d'intégrer différentes techniques formelles de la Logique Combinatoire avec des types (Curry) et sa programmation fonctionnelle (Haskell) avec une théorie énonciative du temps et de l'aspect. Nous proposons des calculs formels de valeurs aspecto-temporelles (processus inaccompli présent, processus inaccompli passé, événement passé et état-résultant présent) associées à des représentations de significations verbales sous forme de schèmes applicatifs.

Abstract This work is in line with the research centered on an approach of the Artificial Intelligence and the Computational linguistic. It allows integrating different formal technologies of the Combinatory Logic with types (Curry) and their functional programme (Haskell) with an enunciative theory of the tense and the aspect. We propose formal calculus of aspecto-temporal values (present unaccomplished process, past unaccomplished process, past event and present resultative-state) associated with representations of verbal meanings in the form of applicative schemes.

Mots-clés : Logique Combinatoire, Référentiel énonciatif, Schème sémantico-cognitif, Grammaire Applicative et Cognitive, Haskell

Keywords: Combinatory Logic, Enunciative Frame of reference, Semantic-cognitive scheme, Applicative and Cognitive Grammar, Haskell

1 Introduction

Le temps et l'aspect sont étroitement liés dans une catégorie grammaticale. Il est donc difficile de relever les marqueurs grammaticaux tels qu'ils soient uniquement temporels ou uniquement aspectuels. Cependant, selon Desclés (1990a), nous pouvons distinguer ces deux notions par commodité : d'une part l'aspect construit un procès à partir d'une relation prédicative atemporelle ; d'autre part le temps renvoie plutôt à un repérage du procès par rapport à l'énonciateur. Autrement dit, l'aspect renvoie à une visée de la relation prédicative que se représente l'énonciateur, et le temps renvoie à des relations qui situent le procès soit dans le référentiel de l'énonciateur soit dans un autre système référentiel. Aucune langue n'a de marqueurs uniquement temporels ou uniquement aspectuels. Nous partons, donc, du principe qu'il n'existe pas de distinction nette entre les marqueurs grammaticaux aspectuels et les marqueurs grammaticaux temporels. Ce sont des valeurs aspecto-temporelles qui présentent à la fois des caractéristiques aspectuelles et temporelles.

Le cadre théorique dans lequel nous développons nos analyses linguistiques est celui de la Grammaire Applicative et Cognitive (Desclés, 1990a ; Desclés et alii., 2010) - désormais GA&C. Ce modèle linguistico-computationnel est polystratal avec 7 niveaux de représentations, dont chaque niveau s'articule entre eux par des rôles d'équivalences entre *definiendum* et *definiens*, pour des transformations de représentations.

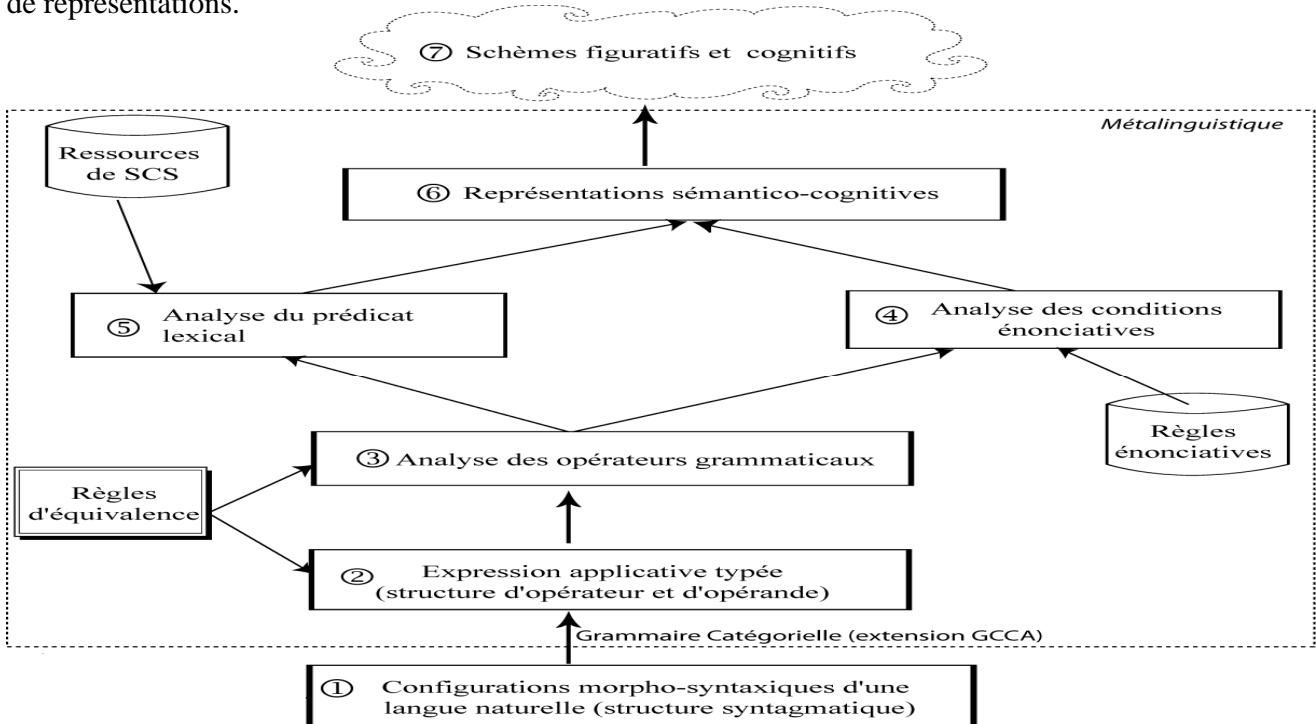


Figure 1 : Sept niveau de l'architecture computationnelle de la GA&C

Ce modèle est analogue à la compilation de programme et implémentable sur des machines informatiques à l'aide de langages de programmation fonctionnelle comme HASKELL, F#, Lisp, etc. Pour analyser certains phénomènes aspecto-temporels et énonciatifs, ce modèle met en œuvre des opérateurs aspecto-temporels qui portent sur une relation prédicative propositionnelle donnant des valeurs aspectuelles (état, événement, processus, ...), et des relations sur des intervalles topologiques (ouverts, fermé, ...). Notre travail diffère de la DRT (Kamp et Reyle, 1993) qui traite aussi le temps et l'aspect; mais comme la DRT ne dispose pas d'opérateur topologique, il lui est difficile de représenter la valeur de l'état résultant et du processus inaccompli.

Ce modèle fait appel à un formalisme unique, la Logique Combinatoire (LC) de Curry (1958, 1972), ce qui amène à représenter chaque énoncé en une expression applicative où les opérateurs de différents types s'appliquent à des opérands. La LC a été utilisée pour la première fois par Shaumyan (1987) pour décrire et analyser les phénomènes linguistiques. Ce type d'approche s'est étendu au domaine de la sémantique cognitive dans la GA&C. La caractéristique fondamentale de la LC est applicative : un opérateur s'applique à un opérande pour fabriquer un résultat. Les combinateurs sont des opérateurs abstraits qui s'appliquent à des opérands pour les composer ou les transformer. Les expressions applicatives constituent des *expressions combinatoires*. Les combinateurs élémentaires se combinent entre eux et construisent des combinateurs plus complexes¹. La signification de chaque combinateur est définie par son action sous forme d'une β -réduction (par des règles d'introduction et d'élimination). La β -réduction vérifie la propriété de Church-Rosser, c'est-à-dire la propriété de confluence : la forme

¹ Les combinateurs dérivés (à puissance et à distance) sont définis à partir de ces combinateurs élémentaires qui étendent la portée d'un combinateur ou agissent à distance. (Curry, 1958, 1972)

INFÉRENCES ASPECTO-TEMPORELLES ANALYSÉES AVEC LA LOGIQUE COMBINATOIRE
réduite, si elle existe, est unique. Donnons quelques combinateurs avec leurs règles d'action² :

B (composition fonctionnelle) : $\mathbf{B} X Y Z \geq_{\beta} X (Y Z)$

C (conversion des deux arguments d'un opérateur binaire) : $\mathbf{C} X Y Z \geq_{\beta} X Z Y$

Notre analyse utilise les concepts de base de la théorie aspecto-temporelle développée dans le courant européen (Benveniste 1966, Comrie 1976, Culioli 1999, Desclés 1995, ...). La distinction de base entre *état*, *événement* et *processus* (Desclés, 1990b) implique la prise en compte d'intervalles topologiques sur lesquels se réalise la relation prédicative. Cette relation aspectualisée (sous la forme d'un état réalisé sur un intervalle ouvert, ou d'un événement réalisé sur un intervalle fermé, ou encore d'un processus réalisé sur un intervalle fermé à gauche et ouvert à droite, car inaccompli) doit être prise en charge par un énonciateur ('JE') sous la forme d'un opérateur d'énonciation (ou opérateur de prise en charge) 'JE-DIS', l'énonciation étant elle-même un processus inaccompli réalisé sur un intervalle 'J⁰', d'où l'opérateur aspectuel 'PROC_{J0}' (processus réalisé sur 'J⁰'). Nous ferons appel aux représentations des significations lexicales sous forme de schèmes sémantico-cognitifs (désormais SSC) qui sont organisés par des λ -expressions typées. Un certain nombre de primitives cognitives générales utilisées dans la construction des schèmes (Desclés, 1990a) sont liées à la perception (primitive relationnelle de mouvement 'MOUVT', primitive de repérage 'REP', primitive d'accessibilité 'ACCS', ...) et à l'action (primitive d'effectuation 'FAIRE', primitive de contrôle par un agent 'CONTR', primitive de visée téléonomique 'TELEO'). Dans cet article, nous allons montrer comment intégrer dans une même approche formelle : (i) les représentations aspecto-temporelles avec un calcul opérant sur les intervalles ; (ii) les représentations des significations lexicales en particulier verbales sous forme de SSC ; (iii) l'utilisation des opérateurs généraux de la LC pour exprimer les représentations et les calculs effectués.

2 Analyse formelle par représentation applicative

Pour illustrer nos propos, partons des exemples suivants et de leurs formules de représentations métalinguistiques associées que nous allons commenter :

(1) processus inaccompli actuel : *[En ce moment,] Luc donne le livre à Jean*

(1') PROC_{J0} ((JE-DIS) (& (PROC_{J1} (donner T³ T² T¹)) ([$\delta(J^1) = \delta(J^0)$])))

(2) processus inaccompli passé : *Luc donnait le livre à Jean [quand Marie est arrivée]*

(2') PROC_{J0} ((JE-DIS) (& (PROC_{J1} (donner T³ T² T¹)) ([$\delta(J^1) < \delta(J^0)$])))

(3) événement passé : *Luc a donné le livre à Jean*

(3') PROC_{J0} ((JE-DIS) (& (EVEN_{F1} (donner T³ T² T¹)) ([$\delta(F^1) < \delta(J^0)$])))

(4) état-résultant : *Luc a donné le livre à Jean, [donc maintenant le livre est la possession de Jean]*

² Un opérateur 'X' qui s'applique à un opérande 'Y' construit un résultat 'Z'. Le programme de l'application de 'X' à 'Y' est noté par une simple juxtaposition (non associative) qui préfixe l'opérateur à son opérande, d'où l'écriture: 'Z = X Y'. Pour simplifier les écritures et en évitant ainsi des parenthèses superflues, nous posons : 'X Y Z =_{def} (X Y) Z' ; il est alors clair que les deux expressions 'X Y Z' et 'X (Y Z)' ne sont pas équivalentes et ne signifient pas le même programme applicatif.

(4') PROC_{J0} ((JE-DIS) (& (ETAT-RESUL_{U1} (donner T³ T² T¹)) ([δ(U¹) = δ(J⁰)])))

Les jeux de coordonnées temporelles expriment des relations de concomitance (par exemple δ(J¹) et δ(J⁰) sont les bornes droites ouvertes de 'J¹' et 'J⁰', [δ(J¹) = δ(J⁰)]³ ou de non-concomitance (par exemple [δ(J¹) < δ(J⁰)]). Les représentations sont exprimées dans un langage applicatif ou fonctionnel où un opérateur 'X' s'applique à un opérande 'Y' en notant cette application par 'X Y'. A chaque exemple, nous associons un diagramme figuratif et formel et le nom de la valeur aspectuelle correspondante.

Diagrammes figuratifs de chaque énoncé	Noms des valeurs aspectuelles
<p>1). <i>En ce moment, Luc donne le livre à Jean.</i></p>	Processus inaccompli actuel
<p>2). <i>Luc donnait le livre à Jean quand Marie est arrivée.</i></p>	Processus inaccompli translaté (ou processus inaccompli passé)
<p>3). <i>Luc a donné le livre à Jean.</i></p>	Événement passé (ou événement accompli passé)
<p>4). <i>Luc a donné le livre à Jean, donc maintenant le livre est la possession de Jean.</i></p>	État-résultant (ou état accompli présent)

Tableau 1 : Diagrammes figuratifs et formels

Les représentations figurales, sous forme de diagrammes expriment bien les relations temporelles entre les intervalles de réalisation des relations prédicatives (processus inaccompli 'J¹' ou événement 'F¹') et l'intervalle de 'J⁰' (processus énonciatif) de la prise en charge énonciative par l'énonciateur. L'intervalle 'U¹' dans l'énoncé (4) est un intervalle de réalisation de valeur aspectuelle de l'état-résultant de l'événement passé (réalisé sur 'F¹'); 'U¹' est un intervalle ouvert qui est immédiatement contigu à l'intervalle fermé 'F¹'; cet état-résultant réalisé sur 'U¹' correspond à l'état acquis « *Luc a acquis une nouvelle propriété (avoir-donné le-livre à Jean)* », c'est-à-dire que l'intervalle 'U¹' a sa borne droite qui coïncide avec la borne droite de 'δ(J⁰)' [δ(U¹)=δ(J⁰)]; l'état-résultant est un « *présent* » (en anglais, *present perfect*) et non pas un passé (voir Benveniste 1966, Desclés et Guentchéva 1995).

La théorie représentative de Reichenbach (1947), reprise par plusieurs linguistes, opère avec des instants alors que notre approche opère avec des intervalles d'instant, ce qui permet de rendre compte des oppositions aspectuelles entre (i) les inaccomplis (état ou processus) réalisés sur un intervalle ouvert ou semi-ouvert avec une borne droite ouverte (indiquant la non prise en compte d'un dernier instant de réalisation) et (ii) les événements accomplis réalisés sur un intervalle fermé (pas nécessairement ponctuel) avec la prise en compte d'un dernier instant de réalisation.

³ Nous noterons les bornes droite et gauche d'un intervalle quelconque 'I' par respectivement 'δ(I)' et 'γ(I)'. Elles peuvent être ouvertes ou fermées selon qu'elles n'appartiennent ou n'appartiennent pas à l'intervalle 'I'.

2.1 Représentation de la signification du lexique verbal

La représentation de la signification de « donner » a été largement étudiée en science cognitive (par exemple, « to give » Langacker 1991, Pottier 2000), et dans le cas de la GA&C, ce verbe a été analysé en français, en coréen et en russe (Son 2006, Ivanova-Tarasova 2009). La représentation de la signification relative à notre exemple est donnée par la λ -expression applicative du SSC:

$$\lambda z. \lambda y. \lambda x. (\text{CONTR} (\text{FAIRE} (\text{MOUVT}_{F12} (\text{SIT}^1_{O1} [x,y,z]) (\text{SIT}^2_{O2} [x,y,z])) x) x)$$

avec: $\text{SIT}^1_{O1} [x,y,z] = \text{ET} (\text{ACCS } x \ y) (\text{NOT} (\text{ACCS } z \ y))$

$$\text{SIT}^2_{O2} [x,y,z] = \text{ET} (\text{ACCS } z \ y) (\text{NOT} (\text{ACCS } x \ y))$$

L'expression au-dessus signifie (voir aussi la figure 2) : l'agent 'x' exerce la capacité de mouvement d'une situation 'SIT¹_{O1}' à une autre situation 'SIT²_{O2}'; ce mouvement est effectué et réalisé pendant l'intervalle fermé 'F¹²', des instants de l'événement du mouvement ('MOUVT_{F12}') entre deux situations ; dans la première situation 'SIT¹_{O1}', l'entité 'y' est accessible à l'entité 'x' (ACCS x y)⁴, et non pas à l'entité 'z' ; cette situation est réalisée (est vraie) durant l'intervalle de validation d'état 'O¹' ; dans la deuxième situation 'SIT²_{O2}', l'entité 'y' est accessible à 'z' mais cette fois, n'est pas accessible à 'x' ; cette situation correspond à l'intervalle de validation d'un autre état 'O²'. Ainsi, le mouvement de l'accessibilité d'une entité 'y' est effectué en passant d'une situation à une autre, et tout ce passage n'est pas lié à l'énonciation. Le schème décrit représente la signification du prédicat « donner »⁵ hors de toute prise en charge par un énonciateur ; ce prédicat étant événementiel, nous pouvons poser⁶ :

$$\text{donner } A^3 A^2 A^1 = (\exists \text{ intervalle } F^{12}) \text{ EVEN}_{F12} (\text{donner } A^3 A^2 A^1)$$

Le problème auquel nous sommes confrontés est le suivant : *Comment le schème (sous forme de λ -expression) peut-il être "réunitarisé" en un prédicat verbal avec trois places d'arguments ?* En effet la prédication verbale dans une langue n'opère pas avec le schème mais avec les unités linguistiques qui sont des prédicats verbaux. La LC est convoquée pour exprimer cette synthèse intégrative. Pour cela, nous partons du schème et nous introduisons successivement un certain nombre de combinateurs de la LC de façon à dégager les places d'arguments et à faire apparaître un opérateur fonctionnel, une certaine composition entre 'X', qui compose entre elles les primitives en un tout intégré. Nous ne donnons pas, ici, la forme exacte de 'X' (car cela exigerait des développements techniques de la LC).

2.2 Prise en charge d'une prédication verbale

Il s'agit maintenant de montrer comment la relation prédicative (*donner* A³ A² A¹) peut être prise en charge par un processus énonciatif réalisé sur 'J⁰'. Peut-on exprimer cela à l'aide d'un calcul formel faisant apparaître l'analyse sémantique des marqueurs morphologiques (marqueurs de temps au sens de « tense ») dans une démarche « top-down » (c'est-à-dire descendante) ? Là encore, la LC nous permet de

⁴ ACCS est un domaine des entités accessibles à une autre entité. 'ACCS x y' : c'est le domaine des entités 'y' qui sont accessibles à l'entité 'x'.

⁵ Nous ne précisons pas, ici, les types fonctionnels des primitives MOUVT, CONTR, FAIRE, etc. et les types des arguments (ici, 'x', 'y', et 'z' ont le même type de 'J', entité individuelle).

⁶ Pour simplifier les écritures, introduisons le symbole 'A' qui désigne les termes nominaux (ou les actants) ; 'A³', 'A²' et 'A¹' associés aux trois syntagmes nominaux respectifs "le-livre", "Jean" et "Luc".

réaliser ces calculs sous forme d'une déduction formelle qui introduit successivement des combinateurs. Avec l'analyse de la signification du prédicat « donner » par son schéma associé et suivant l'analyse d'opération élémentaire d'insertion dans le référentiel énonciatif, nous sommes capables d'expliquer l'inférence : *Luc a donné le livre à Jean → Maintenant, Jean a le livre et Luc n'a plus de livre*

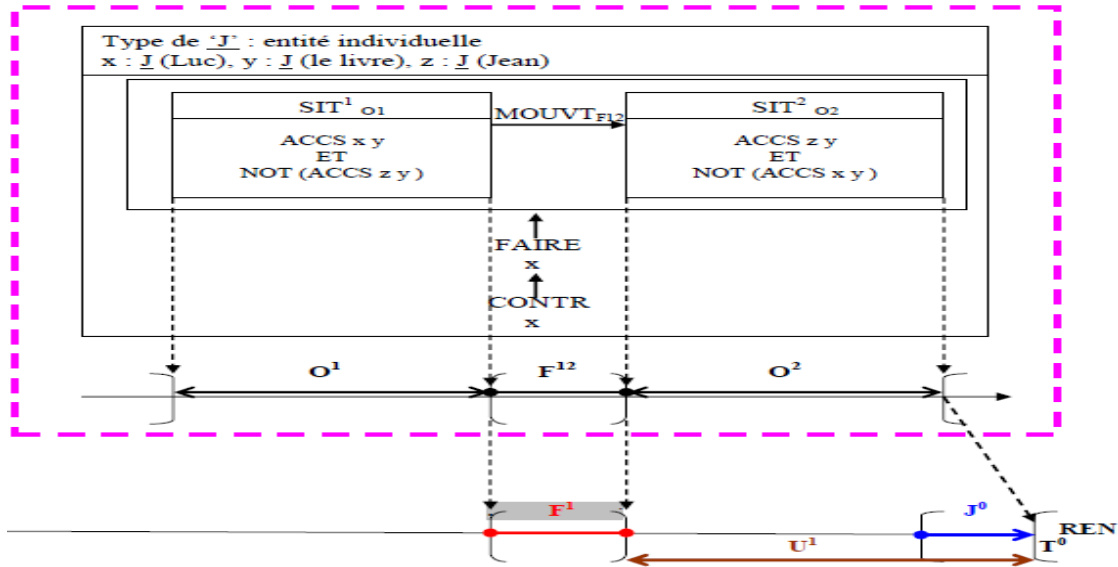
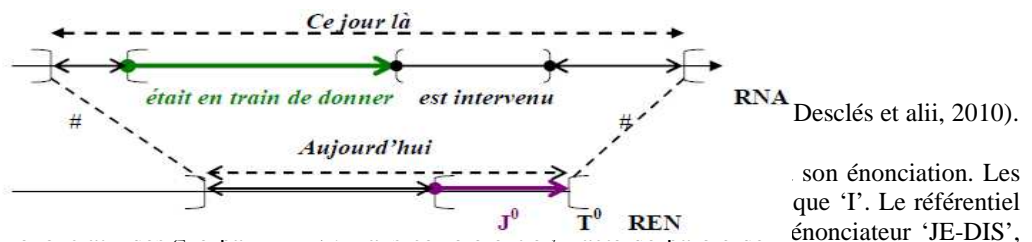


Figure 2 : SSC de « donner » avec projection sur le référentiel énonciatif (REN)

En énonçant l'événement « *Luc a donné le livre à Jean* » nous effectuons les relations temporelles : $[F^1 = F^{12}]$ et $[\delta(F^1) < \delta(J^0)]$; l'état-résultant présent « [*Maintenant*] *Jean a le livre* » et le résultatif « *Luc n'a plus de livre* » engendrés par la réalisation de l'événement durant l'intervalle 'F¹' conduit aux relations temporelles : $[\delta(F^1) = \gamma(U^1)]$ et $[\delta(U^1) = \delta(J^0)]$ ⁷. Dans le même esprit, nous devons complexifier le calcul pour représenter l'ensemble des valeurs aspecto-temporelles du français en utilisant la notion de *référentiel temporel* (référentiel énonciatif (REN), référentiel non-actualisé (RNA), référentiel des possibles, ...), et tenir compte des différents référentiels (Desclés et Guentchéva, 2006) où la relation prédicative peut venir s'insérer⁸. En effet, l'imparfait comme le passé composé peuvent fonctionner dans des référentiels temporels qui ne sont pas nécessairement liés à l'énonciation. Par exemple : « *Un jour, Luc a donné le livre de géographie à Paul et c'est cela qui a déclenché la vocation de Paul : aller explorer les terres de la planète* ». Le marqueur 'un jour' est l'indication d'un changement de référentiel puisque la narration évoquée ne peut plus être repérée par rapport à son énonciation. De même, dans une narration : « *Ce jour-là, Paul était en train de donner sa propriété à son neveu lorsque son fils est intervenu pour ...* » Là encore le marqueur 'ce jour-là' indique une narration autonome et indépendante de son énonciation. Cette analyse peut être représentée à l'aide de la figure 3 :



⁷ Nous ne donnons pas ici

⁸ Chaque référentiel s'or...

Desclés et alii, 2010).

son énonciation. Les que 'T'. Le référentiel énonciateur 'JE-DIS',

de base est le référenti... sous forme de processus, repère toutes les relations prédicatives des énoncés. Certains marqueurs (*Ce jour-là*, *Un jour*, *l'orage éclata*, *si tu fais...*) permettent de ne pas repérer à **REN** interne et créent un autre référentiel où les relations prédicatives qui les contiennent y sont insérées.

Figure 3 : Relation entre référentiels temporels

Dans ce diagramme 3, la relation ‘#’ exprime un changement de référentiel entre REN et RNA. Dans le référentiel non-actualisé (RNA), tous les états, tous les événements et tous les processus sont repérés les uns par rapport aux autres et non par rapport à ‘J⁰’. Le changement de référentiel permet de prendre en compte l’opposition de Benveniste entre « discours » et « histoire » bien que Benveniste n’introduise pas la notion de référentiel temporel. Il s’agit donc d’intégrer cette dernière notion dans un système de changement de référentiel et les calculs formels qui s’y attachent. C’est là l’objet de notre travail.

Le travail formel nous a amené à introduire les relations temporelles dans la portée de l’opérateur de prise en charge ‘JE-DIS’, ce qui n’était pas le cas dans les articles de Desclés (2004, 2005), qui analysait (1’) et (2’) par :

$$(1'') \text{ (PROC}_{J_0} \text{ ((JE-DIS) (PROC}_{J_1} \text{ (donner } T^3 T^2 T^1 \text{))))} \ \& \ ([\delta(J^1) = \delta(J^0)])$$

$$(2'') \text{ (PROC}_{J_0} \text{ ((JE-DIS) (PROC}_{J_1} \text{ (donner } T^3 T^2 T^1 \text{))))} \ \& \ ([\delta(J^1) < \delta(J^0)])$$

La représentation que nous proposons est plus “naturelle” puisque la relation temporelle est bien un constituant de « ce qui est dit ».

3 Traitement informatique de l’analyse aspecto-temporelle

Si l’être humain a la capacité d’effectuer des inférences aspecto-temporelles, une machine doit être guidée par des règles explicites pour obtenir des résultats analogues. Il faut pour cela évaluer à chaque étape de calcul la structure de l’expression et ses composantes logiques et sémantiques. Du point de vue computationnel, il s’agit d’apporter la preuve que les analyses précédentes sont effectives et peuvent être réalisées automatiquement, ce qui permettra des applications ultérieures. L’utilisation de l’informatique dans un mode expérimental permettra de tester la force explicative et opératoire de la théorie aspecto-temporelle et de la GA&C. La programmation fonctionnelle est un paradigme de programmation dans lequel les notions de fonction et de récursivité tiennent une place essentielle, notamment à travers l’importance de l’appel récursif de fonctions dans le programme. En particulier, l’action des combinateurs, conçus comme des opérateurs de composition, d’opérateur plus élémentaire, qui est facilement réalisable et gérable en programmation fonctionnelle. Pour faire des calculs automatiquement, nous devons fournir à la machine toutes les informations nécessaires et adéquates afin de réduire les combinateurs par étapes. Pour cela, nous avons effectué une première réalisation informatique avec, entre autres, le langage fonctionnel F# (Ro, 2008), puis, avec Haskell. Le programme, implémenté à l’aide de B.Djioua, prend en entrée une phrase comme « *Luc a donné le livre à Jean* » et renvoie sa représentation sémantique sous la forme d’un schème instancié. Le programme procède à une première analyse syntaxique dans le cadre des Grammaires Catégorielles construisant ainsi une expression applicative associée (Biskri et Desclés, 2005); ensuite le programme construit les conditions d’énonciation et parallèlement engendre une représentation sémantico-cognitive à partir de la décomposition de prédicat « *donner* » à l’aide de l’opérateur ‘X’. Le programme termine par une intégration des conditions énonciatives et du schème instancié donnant l’analyse sémantique de la phrase présentée en entrée. Ainsi, nous pouvons rendre compte automatiquement de certaines inférences comme celle que nous avons vu au §2.2. Le travail est en cours pour traiter un plus grand nombre d’exemples. Dans cet article, nous montrons uniquement une partie de l’implémentation, relative aux opérations

aspecto-temporelles. Plus précisément, nous avons réalisé un programme de reconnaissance chargé de reconstruire les conditions aspecto-temporelles à partir des marqueurs morphologiques.

3.1 Combinateurs et opérateurs aspecto-temporels

La première partie de l'implémentation consiste à définir les combinateurs élémentaires et dérivés (puissance et à distance) de la LC dans le programme. La réduction que nous allons utiliser traite du combinateur effectuable le plus à gauche dans l'expression applicative considérée. Les schémas de types combinateurs sont fonctions de types variables que nous déterminons par l'opération d'unification (Les types 't1', 't2' et 't3'... sont des types variables, ils sont identifiés par unification). Nous montrons ici les définitions de quelques combinateurs et leurs schémas de types (fenêtre à gauche – définitions des combinateurs dans un éditeur de code, fenêtre à droite – schémas de types des combinateurs définis en Haskell via l'interpréteur interactif 'GHCi'):

```

C:\Documents and Settings\hro\Bureau\RECITAL\Haskell\Recital_Haskell.hs - Notepad++
Fichier Edition Recherche Affichage Format Langage Paramétrage Macro Exécution TextFX Plugins Document ?
Recita2.hs Recital_Haskell.hs
1 --Combinateurs élémentaires : B, C et C*-----
2 combiB f g x= f (g x)
3 combiC f x y = f y x
4 combiCC x f = f x -- combinateur C*
5
6 -- Algèbres des combinateurs -----
7 --1 : Puissance des combinateurs, exemple de combinateur 'B' (Solution "ad hoc")-----
8 puissB2 = combiB combiB combiB
9 puissB3 = combiB combiB (combiB combiB combiB)
10 puissB4 = combiB combiB (combiB combiB (combiB combiB combiB))
11 puissB7 = combiB combiB (combiB combiB (combiB combiB (combiB combiB (combiB combiB (combiB combiB combiB))))))
12
13 --2 : Combinateurs à distance , exemple de combinateur 'C' -----
14 distanceC1 = combiB combiC
15 distanceC2 = combiB combiB combiB combiC
16 distanceC3 = combiB combiB (combiB combiB combiB) combiC
17
18 distanceB6 = combiB (combiB (combiB (combiB (combiB (combiB (combiB combiB))))))
19 distanceB8 = combiB (combiB (combiB (combiB (combiB (combiB (combiB (combiB (combiB combiB)))))))
20
GHCi
Loading package ghc-prim ... linking ... done.
Loading package integer ... linking ... done.
Loading package base ... linking ... done.
Prelude> :cd C:\Documents and Settings\hro\Bureau\RECITAL\Haskell
Prelude> :set prompt "Hee-Jin>"
Hee-Jin> :l Recital_Haskell.hs
[1 of 1] Compiling Main                 ( Recital_Haskell.hs, interpreted )
Ok, modules loaded: Main.
Hee-Jin>:t combiB
combiB :: (t1 -> t2) -> (t -> t1) -> t -> t2
Hee-Jin>:t combiC
combiC :: (t1 -> t -> t2) -> t -> t1 -> t2
Hee-Jin>:t puissB2
puissB2 :: (t1 -> t2) -> (t -> t3 -> t1) -> t -> t3 -> t2
Hee-Jin>:t puissB3
puissB3
:: (t2 -> t3) -> (t -> t1 -> t4 -> t2) -> t -> t1 -> t4 -> t3
Hee-Jin>:t distanceC1
distanceC1 :: (t3 -> t -> t1 -> t2) -> t3 -> t1 -> t -> t2
Hee-Jin>:t distanceC2
distanceC2
:: (t -> t1 -> t2 -> t3 -> t4) -> t -> t1 -> t3 -> t2 -> t4
Hee-Jin>:t distanceB6
distanceB6
:: (t8 -> t7 -> t6 -> t5 -> t4 -> t3 -> t -> t1)
-> t8
-> t7
-> t6
-> t5
-> t4
-> t3
-> (t2 -> t)
-> t2
-> t1
Hee-Jin>:t distanceB8
distanceB8
:: (t10 -> t9 -> t8 -> t7 -> t6 -> t5 -> t4 -> t3 -> t -> t1)
-> t10
-> t9
-> t8
-> t7
-> t6
-> t5
-> t4
-> t3
-> (t2 -> t)
-> t2
-> t1
Hee-Jin>

```

Implémentons maintenant quelques opérateurs aspecto-temporels⁹ ainsi que les relations temporelles, introduits au §2.1., de l'événement dans le passé (énoncé (3)) en Haskell. Nous définissons d'abord les types d'entités individuelles (J), de prédicat (Predicat), d'opérateurs aspecto-temporels (Operateurs), d'intervalle, d'instant, en utilisant Haskell *data keyword* (*data constructors* ou *value constructors*) et *recursive types*; ensuite nous définissons les fonctions (*local functions*) pour l'exécution. Le résultat, après exécution, est l'expression applicative de la représentation formelle de l'analyse aspecto-temporelle (voir la formule de représentation métalinguistique de l'énoncé 3, alors 3'). Les définitions et les résultats du programme correspondent à chaque étape du calcul formel.

⁹ Pour l'implémentation de toutes les étapes de niveau morphologique au SSC, voir Desclés & al. (2010), pour l'implémentation des SSC, voir Djoua (2000), et pour l'analyse et son implémentation en F#, voir Ro (2008).

INFÉRENCES ASPECTO-TEMPORELLES ANALYSÉES AVEC LA LOGIQUE COMBINATOIRE

```
Rectal2.hs Rectal_Haskell.hs
21 --Luc a donné le livre à Jean -----
22 data J = Luc | Le_livre | Jean deriving Show
23 data Predicat = Donner_a J J J deriving Show
24 data Operateurs = EVENT_PASSE Predicat
25 | ET Operateurs Operateurs
26 | ENONC_J0 Operateurs
27 | EVEN_F1 Operateurs
28 | Inf Instant Instant
29 | Eq Instant Instant
30 | EVEN_F12 Predicat
31 | CONCOM Intervalle Intervalle
32 | R_01_02_F12
33 | JE_DIS Operateurs
34 | PROC_J0 Operateurs
35 deriving Show
36 data Intervalle = J0|F1|J1|F12|01|02 deriving Show
37 data Instant = Delta Intervalle|Gamma Intervalle deriving Show
38
39 avoir_donne_a x = puisB3 EVENT_PASSE Donner_a x
40 --Exécution : Hee-Jin>avoir_donne_a Jean Le_livre Luc
41 --          EVENT_PASSE (Donner_a Jean Le_livre Luc)
42 r01_02_F12 = (ET (Eq (Delta 01) (Gamma F12)) (Eq (Gamma 02) (Delta F12)))
43 R_01_02_F12 = r01_02_F12
44 enonc_J0 x = combiB PROC_J0 JE_DIS x
45 ENONC_J0 x = enonc_J0 x
46 event_passe = distanceB8 distanceB6 distanceC3 distanceC3 combiC puisB2 ET
47 | (ET (ET (Inf (Delta F1) (Delta J0)) (CONCOM F1 F12)) (r01_02_F12)) enonc_J0 EVEN_F1 EVEN_F12
48 --Exécution :Hee-Jin> event_passe (Donner_a Jean Le_livre Luc)
49 --PROC_J0 (JE_DIS (ET (EVEN_F1 (EVEN_F12 (Donner_a Jean Le_livre Luc)))
50 --          (ET (ET (Inf (Delta F1) (Delta J0)) (CONCOM F1 F12))
51 --          (ET (Eq (Delta 01) (Gamma F12)) (Eq (Gamma 02) (Delta F12))))))
52 --          )
53 --          )
54 --          )
```

```
GHCI
Hee-Jin>
Hee-Jin>avoir_donne_a Jean Le_livre Luc
EVENT_PASSE (Donner_a Jean Le_livre Luc)
Hee-Jin>
Hee-Jin>
Hee-Jin>event_passe (Donner_a Jean Le_livre Luc)
PROC_J0 (JE_DIS (ET (EVEN_F1 (EVEN_F12 (Donner_a Jean Le_livre L
nf (Delta F1) (Delta J0)) (CONCOM F1 F12)) (ET (Eq (Delta 01) (G
Gamma 02) (Delta F12))))))
Hee-Jin>
```

4 Conclusion

L'utilisation de la Logique Combinatoire que nous venons de montrer sur les quelques exemples permet, d'une part, d'articuler des niveaux de représentations de la GA&C entre eux dans un processus de changement de niveau au sein d'une même architecture computationnelle, et d'autre part, de synthétiser un prédicat à partir de l'analyse de sa représentation sémantique. Il existe de nombreuses analyses formelles automatiques de la morphologie et de la syntaxe, mais les constructions automatiques de représentation sémantique sont plus complexes à décrire et à formaliser. Les réalisations actuelles sont souvent des applications très limitées à un domaine externe parfaitement circonscrit. De plus, peu de réalisations effectives portent sur la prise en compte de la sémantique du temps et de l'aspect, excepté les études d'Asher et alii. (1995) qui sont basées sur une approche de l'aspectualisation et Kamp (1993). C'est ainsi que notre recherche permettrait la réalisation d'un programme d'automatisation de l'analyse aspecto-temporelle. Nous souhaiterions présenter ultérieurement comment certaines inférences aspecto-temporelles, qui mettent en jeu des significations lexicales et les conditions aspecto-temporelles peuvent être traitées par des calculs formels et réalisées par des programmes informatiques.

Références

ASHER N., AURNAGUE M., BRAS M., SABLAYROLLES P., VIEU L. (1995). "De l'espace-temps dans l'analyse du discours". *Semiotique*, 9:11–62.

BENVENISTE É. (1966, 1974). *Problèmes de linguistique générale* 1 & 2. Paris: Gallimard.

BISKRI I., DESCLÉS J-P. (2005). "Applicative and Combinatory Categorical Grammar and Subordinate Constructions in French". *International Journal on Artificial Intelligence Tools*, v14. N°1 & 2, 125-136.

COMRIE B. (1976). *Aspect, an Introduction to the Study of Verbal Aspect and Related Problems*. London: Cambridge University Press.

CULIOLI A. (1999). *Pour une linguistique de l'énonciation, Formalisation et opérations de repérage*. Tome 2, Paris: Ophrys.

HEE-JIN RO

- CURRY H.B., (1958, 1972). *Combinatory Logic*. Volume 1 & 2. Amsterdam: North-Holland Publishing.
- DESCLÉS J.-P., (1990a). *Langages applicatifs, langues naturelles et cognition*. Paris: Hermès.
- DESCLÉS J.-P., (1990b). "State, Event, Process, and Topology". in: *General Linguistics*, 29, 3:159-200. University Park and London: Pennsylvania State University Press.
- DESCLÉS J.-P., GUENTCHÉVA Z. (1995). "Is the notion of Process necessary? A fundamental distinction process in progress and state activity". in: *Temporal Reference, Aspect and Actionality*, P.-M Bertinetto; V. Bianchi; J. Higginbotham; M. Squartini (eds.), Torino: Rosenberg and Sellier, 1995, p55-70.
- DESCLÉS J.-P. (2004). "Combinatory Logic, Language and Cognitive Representations". in Weingarten P. (ed.), *Alternative Logics. Do Sciences Need Them?*, Springer, 115-148.
- DESCLÉS J.-P. (2005). "Reasoning and Aspectual-Temporal Calculus". in: Vanderveken D. (eds.), *Logic, Thought and Action*, Springer, pp. 217-244.
- DESCLÉS J.-P., GUENTCHÉVA Z. (2006). "Référentiels aspecto-temporels dans les textes". in: *Etudes Cognitives*, 7. SOW, Warszawa. pp. 6-38.
- DESCLÉS J.-P., DJIOUA B., RO H.-J. (2010). "Computational Approach of Cognitive Semantic by means of Combinatory Logic". soumis à publication.
- DJIOUA B. (2000). *Modélisation Informatique d'une base de connaissances lexicales (DiSSC) : Réseaux polysémiques et Schèmes Sémantico-Cognitifs*, Thèse de doctorat, Université Paris-Sorbonne.
- HINDLEY, J.R., SELDIN, J.P. (1986). *Introduction to Combinators and λ -Calculus*. Cambridge Univ. Press.
- IVANOVA-TARASOVA E. (2009). *Approche cognitive et formelle de la polysémie verbale : les verbes de transfert en français et en russe*, Thèse de doctorat, Université Paris-Sorbonne.
- KAMP H., REYLE U. (1993). *From discourse to logic*. Kluwer Academic Publisher.
- LANGACKER R.W. (1989, 1991). *Foundations of cognitive grammar*, volume 1, *Theoretical prerequisites*, volume 2, *Description Application*, Stanford University Press
- O'SULLIVAN B., GOERZEN J., STEWART D. (2008). *Real world Haskell*. Sebastopol : O'Reilly
- POTTIER B. (2000). *Représentations mentales et catégorisations linguistiques*, Louvain, Paris, Peeters.
- REICHENBACH H. (1947). *Elements of Symbolic Logic*. New York, reprinted in J.M. Moravcsik (eds.), 1974, :Logic and Philosophy for Linguists, A Book of readings, The Hague: Mouton, pp. 122-141.
- RO, H.-J., (2008). *Traitement automatique de l'analyse d'inférences aspecto-temporelles : modélisation logique et informatique*. Mémoire de Master de Recherche. Université Paris-Sorbonne.
- SHAUMYAN S.K. (1987). *A Semiotic Theory of Natural Languages*. Bloomington: Indiana Univ Press.
- SON H. (2006). *Représentation sémantico-cognitive des verbes de transfert*, Thèse de doctorat, Université Paris-Sorbonne.