

Type-shifting

Laurent Roussarie

Sémantique, M1 LTD

2013

Les DP sont de type $\langle\langle e, t \rangle, t\rangle$

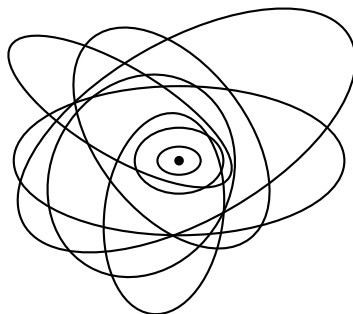
Montague (1973)

Tous les DP sont de type $\langle\langle e, t \rangle, t\rangle$.

Les DP sont de type $\langle\langle e, t \rangle, t\rangle$

Montague (1973)

Tous les DP sont de type $\langle\langle e, t \rangle, t\rangle$.



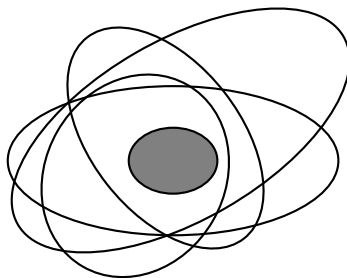
$\lambda P[P(a)]$

Alice

Les DP sont de type $\langle\langle e, t \rangle, t\rangle$

Montague (1973)

Tous les DP sont de type $\langle\langle e, t \rangle, t\rangle$.

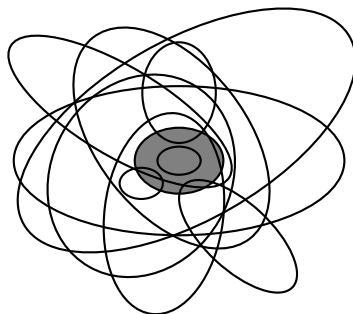


$\lambda P \forall x [\text{enfant}(x) \rightarrow [P(x)]]$
tous les enfants

Les DP sont de type $\langle\langle e, t \rangle, t\rangle$

Montague (1973)

Tous les DP sont de type $\langle\langle e, t \rangle, t\rangle$.



$\lambda P \exists x [\text{enfant}(x) \wedge [P(x)]]$

Un enfant

Pourquoi faire compliqué ?

Argument des conjonctions : on peut conjoindre (avec *et*, *ou*) deux DP s'ils ont le même type.

(1) Blanche-Neige et chaque nain...

*	e	<i>et</i>	$\langle\langle e, t \rangle, t\rangle$
	$\langle\langle e, t \rangle, t\rangle$	<i>et</i>	$\langle\langle e, t \rangle, t\rangle$

Pourquoi faire compliqué ?

... quand on peut faire plus simple

- Partee and Rooth (1983) : Les DP reçoivent lexicalement le type le plus simple ; puis ce type peut être complexifié si l'environnement syntaxique l'exige.
- Partee (1987) : Les DP peuvent « voyager » d'un type à l'autre, parmi e , $\langle e, t \rangle$, $\langle \langle e, t \rangle, t \rangle$, selon des règles précises et linguistiquement naturelles.

Le type-shifting

Principe du *type-shifting*

Une expression X de la langue initialement traduite dans LO par l'expression α de type τ_1 est analysée en α' de type τ_2 ($\neq \tau_1$).

Le type-shifting

Principe du *type-shifting*

Une expression X de la langue initialement traduite dans LO par l'expression α de type τ_1 est analysée en α' de type τ_2 ($\neq \tau_1$).

Comment passer d'un type à l'autre ? $\tau_1 \rightarrow \tau_2$

Le type-shifting

Principe du *type-shifting*

Une expression X de la langue initialement traduite dans LO par l'expression α de type τ_1 est analysée en α' de type τ_2 ($\neq \tau_1$).

Comment passer d'un type à l'autre ? $\tau_1 \rightarrow \tau_2$

Proposition

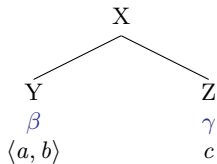
Une opération de changement de type peut être effectuée par une *fonction*, donc un λ -terme, dont le type est $\langle \tau_1, \tau_2 \rangle$.

Et cette fonction est activée par la structure syntaxique, en particulier s'il y a *mismatch* de types dans la composition.

Changement de type de Z dans $[_X Y Z]$

Illustration

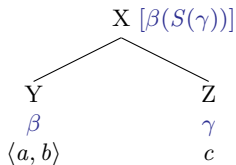
S est la fonction qui change le type de γ



Changement de type de Z dans $[_X Y Z]$

Illustration

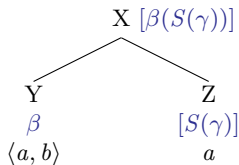
S est la fonction qui change le type de γ



Changement de type de Z dans $[_X Y Z]$

Illustration

S est la fonction qui change le type de γ

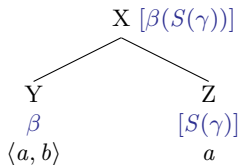


Avec S de type $\langle c, a \rangle$. Et donc $[S(\gamma)]$ est de type a .

Changement de type de Z dans $[_X Y Z]$

Illustration

S est la fonction qui change le type de γ



Avec S de type $\langle c, a \rangle$. Et donc $[S(\gamma)]$ est de type a .

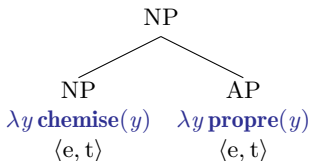
$$\begin{array}{lcl}
 \text{Syn :} & X & \rightarrow Y \quad Z \\
 & \downarrow & \downarrow \quad \downarrow \\
 \text{Sem :} & [\beta(S(\gamma))] & \leftarrow \beta \quad \gamma \\
 & \text{ou } [S(\gamma)](\beta) &
 \end{array}$$

Type-shifting

Application à la promotion des adjoints

- $\mathbf{PrAd} = \lambda P \lambda Q \lambda x [[Q(x)] \wedge [P(x)]]$
avec $P, Q \in \mathcal{V}ar_{\langle e, t \rangle}$

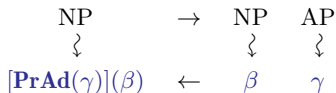
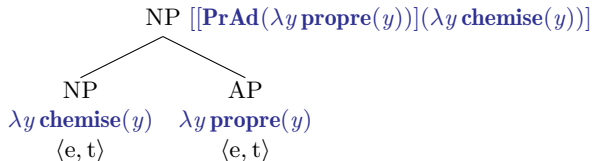
de type $\langle \langle e, t \rangle, \langle \langle e, t \rangle, \langle e, t \rangle \rangle \rangle$



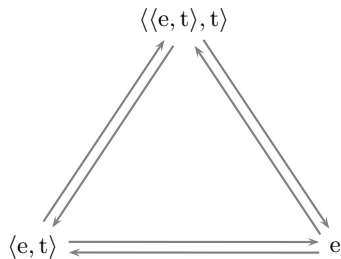
Type-shifting

Application à la promotion des adjoints

- $\mathbf{PrAd} = \lambda P \lambda Q \lambda x [[Q(x)] \wedge [P(x)]]$ de type $\langle \langle e, t \rangle, \langle \langle e, t \rangle, \langle e, t \rangle \rangle \rangle$
avec $P, Q \in \mathcal{Var}_{\langle e, t \rangle}$



Type-shifting chez les DP



De e à $\langle\langle e, t \rangle, t\rangle$ $e \rightarrow \langle\langle e, t \rangle, t\rangle$

- e est le type dit *référentiel* : noms propre, pronoms personnels, descriptions définies.
- $\langle\langle e, t \rangle, t\rangle$ est le type dit *quantificationnel* : DP quantificationnels.
- Fonction de *montée de type* : **lift** = $\lambda x \lambda P[P(x)]$ de type $\langle e, \langle\langle e, t \rangle, t\rangle \rangle$.

De e à $\langle\langle e, t \rangle, t\rangle$ $e \rightarrow \langle\langle e, t \rangle, t\rangle$

- e est le type dit *référentiel* : noms propre, pronoms personnels, descriptions définies.
- $\langle\langle e, t \rangle, t\rangle$ est le type dit *quantificationnel* : DP quantificationnels.
- Fonction de *montée de type* : **lift** = $\lambda x \lambda P[P(x)]$ de type $\langle e, \langle\langle e, t \rangle, t\rangle \rangle$.

lift :

De e à $\langle\langle e, t \rangle, t\rangle$ $e \rightarrow \langle\langle e, t \rangle, t\rangle$

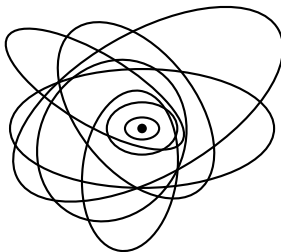
- e est le type dit *référentiel* : noms propre, pronoms personnels, descriptions définies.
- $\langle\langle e, t \rangle, t\rangle$ est le type dit *quantificationnel* : DP quantificationnels.
- Fonction de *montée de type* : **lift** = $\lambda x \lambda P [P(x)]$ de type $\langle e, \langle\langle e, t \rangle, t\rangle \rangle$.

lift :

•

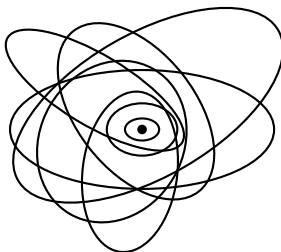
De e à $\langle\langle e, t \rangle, t\rangle$ $e \rightarrow \langle\langle e, t \rangle, t\rangle$

- e est le type dit *référentiel* : noms propre, pronoms personnels, descriptions définies.
- $\langle\langle e, t \rangle, t\rangle$ est le type dit *quantificationnel* : DP quantificationnels.
- Fonction de *montée de type* : **lift** = $\lambda x \lambda P[P(x)]$ de type $\langle e, \langle\langle e, t \rangle, t\rangle \rangle$.

lift :

De e à $\langle\langle e, t \rangle, t\rangle$ $e \rightarrow \langle\langle e, t \rangle, t\rangle$

- e est le type dit *référentiel* : noms propre, pronoms personnels, descriptions définies.
- $\langle\langle e, t \rangle, t\rangle$ est le type dit *quantificationnel* : DP quantificationnels.
- Fonction de *montée de type* : **lift** = $\lambda x \lambda P[P(x)]$ de type $\langle e, \langle\langle e, t \rangle, t\rangle \rangle$.

lift :

Chemin inverse ($\langle\langle e, t \rangle, t\rangle \rightarrow e$) : consiste à passer d'un ensemble d'ensembles à un individu ; quel individu extraire ?

$\langle e, t \rangle \rightarrow e$
Le

Ensemble d'individus $\rightarrow$ individu.

Quel individu prendre ? Plusieurs possibilités.

$$\langle e, t \rangle \rightarrow e$$

Le

Ensemble d'individus $\rightarrow$ individu.

Quel individu prendre ? Plusieurs possibilités.

- Si l'ensemble est un singleton ($\{A\}$), on prend son (unique) élément (A). C'est ce que fait l'opérateur ι . Fabriquons la fonction **iota** ; elle est un moyen de traduire l'article défini singulier.

$$\langle e, t \rangle \rightarrow e$$

Le

Ensemble d'individus $\rightarrow$ individu.

Quel individu prendre ? Plusieurs possibilités.

- Si l'ensemble est un singleton ($\{A\}$), on prend son (unique) élément (A). C'est ce que fait l'opérateur γ . Fabriquons la fonction **iota** ; elle est un moyen de traduire l'article défini singulier.
- $le \rightsquigarrow \mathbf{iota} = \lambda P \gamma x [P(x)]$ de type $\langle \langle e, t \rangle, e \rangle$.

$$\langle e, t \rangle \rightarrow e$$

Le

Ensemble d'individus $\rightarrow$ individu.

Quel individu prendre ? Plusieurs possibilités.

- Si l'ensemble est un singleton ($\{A\}$), on prend son (unique) élément (A). C'est ce que fait l'opérateur γ . Fabriquons la fonction **iota** ; elle est un moyen de traduire l'article défini singulier.
- $le \rightsquigarrow \mathbf{iota} = \lambda P \gamma x [P(x)]$ de type $\langle \langle e, t \rangle, e \rangle$.



$$\langle e, t \rangle \rightarrow e$$

Le

Ensemble d'individus $\rightarrow$ individu.

Quel individu prendre ? Plusieurs possibilités.

- Si l'ensemble est un singleton ($\{A\}$), on prend son (unique) élément (A). C'est ce que fait l'opérateur γ . Fabriquons la fonction **iota** ; elle est un moyen de traduire l'article défini singulier.
- $le \rightsquigarrow \mathbf{iota} = \lambda P \gamma x [P(x)]$ de type $\langle \langle e, t \rangle, e \rangle$.

•

$$\langle e, t \rangle \rightarrow e$$

Le

Ensemble d'individus $\rightarrow$ individu.

Quel individu prendre ? Plusieurs possibilités.

- Si l'ensemble est un singleton ($\{A\}$), on prend son (unique) élément (A). C'est ce que fait l'opérateur γ . Fabriquons la fonction **iota** ; elle est un moyen de traduire l'article défini singulier.

- $le \rightsquigarrow \mathbf{iota} = \lambda P \gamma x [P(x)]$ de type $\langle \langle e, t \rangle, e \rangle$.

•

Que vaut **lift**(**iota**(**roi**)) ?

$\langle e, t \rangle \rightarrow e$

Choix

- Si l'ensemble est non vide, on peut prendre... n'importe quel élément. Une fonction qui fait cela s'appelle une *fonction de choix*. Elle est de type $\langle \langle e, t \rangle, e \rangle$.

$\langle e, t \rangle \rightarrow e$

Choix

- Si l'ensemble est non vide, on peut prendre... n'importe quel élément. Une fonction qui fait cela s'appelle une *fonction de choix*. Elle est de type $\langle \langle e, t \rangle, e \rangle$.
- $\llbracket \text{choix}(P) \rrbracket^{\mathcal{M}, w, g} = D$ ssi $\llbracket P \rrbracket^{\mathcal{M}, w, g} \neq \emptyset$ et $D \in \llbracket P \rrbracket^{\mathcal{M}, w, g}$,
i.e. s'il y a un individu D qui vérifie P .

De et vers $\langle e, t \rangle$ $\langle e, t \rangle \rightarrow e$

Choix

- Si l'ensemble est non vide, on peut prendre... n'importe quel élément. Une fonction qui fait cela s'appelle une *fonction de choix*. Elle est de type $\langle \langle e, t \rangle, e \rangle$.
- $\llbracket \text{choix}(P) \rrbracket^{\mathcal{M}, w, g} = D$ ssi $\llbracket P \rrbracket^{\mathcal{M}, w, g} \neq \emptyset$ et $D \in \llbracket P \rrbracket^{\mathcal{M}, w, g}$, i.e. s'il y a un individu D qui vérifie P .



$\langle e, t \rangle \rightarrow e$

Choix

- Si l'ensemble est non vide, on peut prendre... n'importe quel élément. Une fonction qui fait cela s'appelle une *fonction de choix*. Elle est de type $\langle \langle e, t \rangle, e \rangle$.
- $\llbracket \text{choix}(P) \rrbracket^{\mathcal{M}, w, g} = D$ ssi $\llbracket P \rrbracket^{\mathcal{M}, w, g} \neq \emptyset$ et $D \in \llbracket P \rrbracket^{\mathcal{M}, w, g}$, i.e. s'il y a un individu D qui vérifie P .



De et vers $\langle e, t \rangle$ $\langle e, t \rangle \rightarrow e$

Choix

- Si l'ensemble est non vide, on peut prendre... n'importe quel élément. Une fonction qui fait cela s'appelle une *fonction de choix*. Elle est de type $\langle \langle e, t \rangle, e \rangle$.
- $\llbracket \text{choix}(P) \rrbracket^{\mathcal{M}, w, g} = D$ ssi $\llbracket P \rrbracket^{\mathcal{M}, w, g} \neq \emptyset$ et $D \in \llbracket P \rrbracket^{\mathcal{M}, w, g}$,
i.e. s'il y a un individu D qui vérifie P .



$\langle e, t \rangle \rightarrow e$

à suivre...

- ... On verra d'autres façons régulières de passer d'un ensemble à un « individu ».

De et vers $\langle e, t \rangle$ $\langle e, t \rangle \rightarrow e$

à suivre...

- ... On verra d'autres façons régulières de passer d'un ensemble à un « individu ».



$\langle e, t \rangle \rightarrow e$

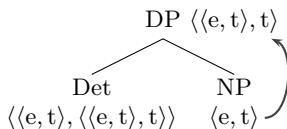
à suivre...

- ... On verra d'autres façons régulières de passer d'un ensemble à un « individu ».



De et vers $\langle e, t \rangle$ $\langle e, t \rangle \rightarrow \langle \langle e, t \rangle, t \rangle$

Le job des déterminants

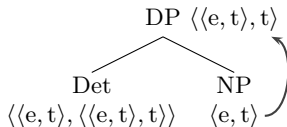
Ensemble $\rightarrow$ ensemble d'ensembles

- $un \rightsquigarrow \lambda P \lambda Q \exists x [[P(x)] \wedge [Q(x)]]$ de type $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$.
- $tout \rightsquigarrow \lambda P \lambda Q \forall x [[P(x)] \rightarrow [Q(x)]]$ de type $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$.

Mais ce n'est pas du type-shifting.

De et vers $\langle e, t \rangle$ $\langle e, t \rangle \rightarrow \langle \langle e, t \rangle, t \rangle$

Le job des déterminants

Ensemble $\rightarrow$ ensemble d'ensembles

- $un \rightsquigarrow \lambda P \lambda Q \exists x [[P(x)] \wedge [Q(x)]]$ de type $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$.
- $tout \rightsquigarrow \lambda P \lambda Q \forall x [[P(x)] \rightarrow [Q(x)]]$ de type $\langle \langle e, t \rangle, \langle \langle e, t \rangle, t \rangle \rangle$.

Mais ce n'est pas du type-shifting.

Partee (1987) propose aussi **THE** = $\lambda P \lambda Q \exists x [\forall y [P(y) \leftrightarrow y = x] \wedge Q(x)]$ Pour nous, c'est (quasiment) équivalent à $\lambda P \lambda Q [Q(\iota x P(x))]$

$e \rightarrow \langle e, t \rangle$

Individu $\rightarrow$ ensemble

Un manière simple : fabriquer l'ensemble singleton $\{A\}$ à partir de A .

- **ident** = $\lambda x \lambda y [y = x]$

$e \rightarrow \langle e, t \rangle$

Individu $\rightarrow$ ensemble

Une manière simple : fabriquer l'ensemble singleton $\{A\}$ à partir de A .

- **ident** = $\lambda x \lambda y [y = x]$

•

$e \rightarrow \langle e, t \rangle$

Individu $\rightarrow$ ensemble

Une manière simple : fabriquer l'ensemble singleton $\{A\}$ à partir de A .

- **ident** = $\lambda x \lambda y [y = x]$



De et vers $\langle e, t \rangle$ $\langle \langle e, t \rangle, t \rangle \rightarrow \langle e, t \rangle$

BE

Ensemble d'ensembles d'individus $\rightarrow$ ensemble d'individus.

$\langle \langle e, t \rangle, t \rangle \rightarrow \langle e, t \rangle$
 BE

Ensemble d'ensembles d'individus $\rightarrow$ ensemble d'individus.

Partee propose la fonction **BE** qui cherche tous les singletons de l'ensemble d'ensembles et range leurs contenus dans un ensemble.

$$\bullet \text{ BE} = \lambda Q \lambda x [Q(\lambda y [y = x])]$$

avec $Q \in \mathcal{Var}_{\langle \langle e, t \rangle, t \rangle}$

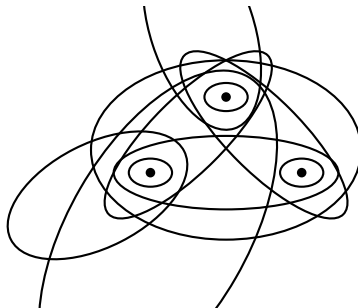
$\langle \langle e, t \rangle, t \rangle \rightarrow \langle e, t \rangle$
BE

Ensemble d'ensembles d'individus $\rightarrow$ ensemble d'individus.

Partee propose la fonction **BE** qui cherche tous les singletons de l'ensemble d'ensembles et range leurs contenus dans un ensemble.

$$\bullet \text{ BE} = \lambda Q \lambda x [Q(\lambda y [y = x])]$$

avec $Q \in \mathcal{Var}_{\langle \langle e, t \rangle, t \rangle}$



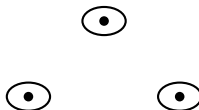
$\langle \langle e, t \rangle, t \rangle \rightarrow \langle e, t \rangle$
BE

Ensemble d'ensembles d'individus $\rightarrow$ ensemble d'individus.

Partee propose la fonction **BE** qui cherche tous les singletons de l'ensemble d'ensembles et range leurs contenus dans un ensemble.

$$\bullet \text{ BE} = \lambda Q \lambda x [Q(\lambda y [y = x])]$$

avec $Q \in \mathcal{Var}_{\langle \langle e, t \rangle, t \rangle}$



$\langle \langle e, t \rangle, t \rangle \rightarrow \langle e, t \rangle$
BE

Ensemble d'ensembles d'individus $\rightarrow$ ensemble d'individus.

Partee propose la fonction **BE** qui cherche tous les singletons de l'ensemble d'ensembles et range leurs contenus dans un ensemble.

$$\bullet \text{ BE} = \lambda Q \lambda x [Q(\lambda y [y = x])]$$

avec $Q \in \mathcal{Var}_{\langle \langle e, t \rangle, t \rangle}$



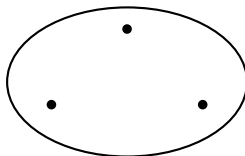
$\langle \langle e, t \rangle, t \rangle \rightarrow \langle e, t \rangle$
BE

Ensemble d'ensembles d'individus $\rightarrow$ ensemble d'individus.

Partee propose la fonction **BE** qui cherche tous les singletons de l'ensemble d'ensembles et range leurs contenus dans un ensemble.

$$\bullet \text{ BE} = \lambda Q \lambda x [Q(\lambda y [y = x])]$$

avec $Q \in \mathcal{Var}_{\langle \langle e, t \rangle, t \rangle}$



De et vers $\langle e, t \rangle$ $\langle \langle e, t \rangle, t \rangle \rightarrow \langle e, t \rangle$

BE

Ainsi $\llbracket \mathbf{BE}(\text{un tigre}) \rrbracket^{\mathcal{M}, w, g} = \llbracket \lambda x \text{ tigre}(x) \rrbracket^{\mathcal{M}, w, g}$.

$\langle \langle e, t \rangle, t \rangle \rightarrow \langle e, t \rangle$
BE

Ainsi $\llbracket \mathbf{BE}(\text{un tigre}) \rrbracket^{\mathcal{M}, w, g} = \llbracket \lambda x \text{tigre}(x) \rrbracket^{\mathcal{M}, w, g}$.

Démonstration :

$$\begin{aligned}
 & \mathbf{BE}(\text{un tigre}) \\
 = & \mathbf{BE}(\lambda P \exists z [\text{tigre}(z) \wedge [P(z)]]) \\
 = & \lambda Q \lambda x [Q(\lambda y [y = x])](\lambda P \exists z [\text{tigre}(z) \wedge [P(z)]]) \\
 = & \lambda x [\lambda P \exists z [\text{tigre}(z) \wedge [P(z)]](\lambda y [y = x])] \\
 = & \lambda x \exists z [\text{tigre}(z) \wedge [\lambda y [y = x](z)]] \\
 = & \lambda x \exists z [\text{tigre}(z) \wedge [z = x]] = \lambda x \text{tigre}(x) \text{ (par définition)}
 \end{aligned}$$

$$\langle \langle e, t \rangle, t \rangle \rightarrow \langle e, t \rangle$$

BE

Ainsi $\llbracket \mathbf{BE}(\text{un tigre}) \rrbracket^{\mathcal{M}, w, g} = \llbracket \lambda x \text{tigre}(x) \rrbracket^{\mathcal{M}, w, g}$.

Démonstration :

$$\begin{aligned}
 & \mathbf{BE}(\text{un tigre}) \\
 = & \mathbf{BE}(\lambda P \exists z [\text{tigre}(z) \wedge [P(z)]]) \\
 = & \lambda Q \lambda x [Q(\lambda y [y = x])](\lambda P \exists z [\text{tigre}(z) \wedge [P(z)]]) \\
 = & \lambda x [\lambda P \exists z [\text{tigre}(z) \wedge [P(z)]](\lambda y [y = x])] \\
 = & \lambda x \exists z [\text{tigre}(z) \wedge [\lambda y [y = x](z)]] \\
 = & \lambda x \exists z [\text{tigre}(z) \wedge [z = x]] = \lambda x \text{tigre}(x) \text{ (par définition)}
 \end{aligned}$$

En revanche $\llbracket \mathbf{BE}(\text{tout tigre}) \rrbracket^{\mathcal{M}, w, g} = \emptyset$, car
 $\llbracket \mathbf{BE}(\text{tout tigre}) \rrbracket = \lambda x \forall y [\text{tigre}(y) \rightarrow [y = x]]$. Or :

(2) *Hobbes est tout/chaque tigre.

Bilan et conclusion

- Le *type-shifting* est un outil très puissant.
Il est à manipuler avec précaution et parcimonie.

Bilan et conclusion

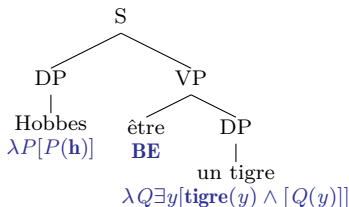
- Le *type-shifting* est un outil très puissant.
Il est à manipuler avec précaution et parcimonie.
- Changement de type = changement de sens ?
Le *type-shifting* peut être vu comme une manière de conditionner l'information sémantique.
Compositionnellement, les contributions sémantiques peuvent être réparties différemment, mais à l'arrivée l'analyse sémantique de la phrase est la même.

Variantes du conditionnement des contributions sémantiques

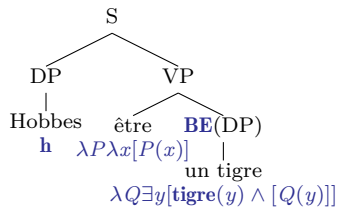
Exemple

(3) Hobbes est un tigre.

Montague (1973) (simplifié) :



Partee (1987) :

Autre possibilité : *un tigre* $\rightsquigarrow \lambda y \text{tigre}(y)$

Classes de DP et types

Corrélation entre propriétés sémantiques des DP et types :

Classes de DP et types

Corrélation entre propriétés sémantiques des DP et types :

- DP référentiels plutôt de type e ;
 - $\lambda x \text{enfant}(x)$

Classes de DP et types

Corrélation entre propriétés sémantiques des DP et types :

- DP référentiels plutôt de type e ;
 - $\iota x \text{ enfant}(x)$
- DP quantificationnels plutôt de type $\langle\langle e, t \rangle, t\rangle$;
 - $\lambda P \forall x [\text{enfant}(x) \rightarrow P(x)]$

Classes de DP et types

Corrélation entre propriétés sémantiques des DP et types :

- DP référentiels plutôt de type e ;
 - $\iota x \text{ enfant}(x)$
- DP quantificationnels plutôt de type $\langle\langle e, t \rangle, t\rangle$;
 - $\lambda P \forall x [\text{enfant}(x) \rightarrow P(x)]$
- DP indéfinis ?

Classes de DP et types

Corrélation entre propriétés sémantiques des DP et types :

- DP référentiels plutôt de type e ;
 - $\iota x \text{ enfant}(x)$
- DP quantificationnels plutôt de type $\langle\langle e, t \rangle, t\rangle$;
 - $\lambda P \forall x [\text{enfant}(x) \rightarrow P(x)]$
- DP indéfinis ? $\langle e, t \rangle$ ou $\langle\langle e, t \rangle, t\rangle$ ou $e \dots$ c'est selon...
 - $\lambda x \text{ enfant}(x)$
 - $\lambda P \exists x [\text{enfant}(x) \wedge P(x)]$
 - $\text{choix}(\lambda x \text{ enfant}(x))$

Référence

- Montague, R. (1973). The proper treatment of quantification in ordinary English. In Hintikka, K. J. J., Moravcsik, J. M. E., and Suppes, P., editors, *Approaches to Natural Language*, pages 221–242. Reidel, Dordrecht.
- Partee, B. (1987). Noun phrase interpretation and type-shifting principle. In Groenendijk, J., de Jongh, D., and Stokhof, M., editors, *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers*, pages 115–144. Foris, Dordrecht.
- Partee, B. and Rooth, M. (1983). Generalized conjunction and type ambiguity. In Bauërle, R., Schwarze, C., and von Stechow, A., editors, *Meaning, Use, and Interpretation of Language*, pages 361–383. Walter de Gruyter, Berlin.