Presupposition and Montague Grammar
(Krahmer 1998, Ch. 5)

Bern Samko
Semantics C (Spring 2010)

Semantic analysis of presuppositions requires partialization of Montague grammar.

- Karttunen & Peters 1979 two-dimensional approach: Assertion and presupposition as two separate expressions of Montague logic.
  - Does not work for sentences with presupposition-quantification interaction:

(1) Somebody managed to succeed George V on the throne of England.

Goal of this chapter: Provide an analysis of a fragment with presupposition-quantification interaction.

- What K&P would have looked like if they had access to Muskens' partialization of Montague Grammar.

1 Background

$\text{TY}_2{}^3$: three-valued two-sorted type theory ($\star$ = undefined expression).

Types:

1. e, s and t are types,
2. if α and β are types, then $(\alpha\beta)$ is a type.

Syntax:

1. If φ and ψ are formulas, then $\neg\varphi$ and $(\varphi \wedge \psi)$ are formulas.
2. If φ is a formula and x is a variable of any type, then $\exists x\varphi$ is a formula.
3. If A is an expression of type $\alpha\beta$ and B is an expression of type α, then (AB) is an expression of type β.
4. If A is an expression of type β and x is a variable of type α, then $\lambda x(A)$ is an expression of type $(\alpha\beta)$.
5. If A and B are expressions of the same type, then $(A \equiv B)$ is a formula.
6. $\star$ is a formula.

Semantics:

1. $[\![\neg\varphi]\!]_g = -[\![\varphi]\!]_g$
2. $[\![\varphi \wedge \psi]\!]_g = [\![\varphi]\!]_g \cap [\![\psi]\!]_g$
3. $[\![\exists x_\alpha \varphi]\!]_g = \bigcup_{d \in D_\alpha} [\![\varphi]\!]_{g[x/d]}$
4. $[\![AB]\!]_g = [\![A]\!]_g([\![B]\!]_g)$
5. $[\![\lambda x_\alpha A]\!]_g =$ the function F s.t. $F(d) = [\![A]\!]_{g[x/d]}$
6. $[\![A \equiv B]\!]_g =$ T iff $[\![A]\!]_g = [\![B]\!]_g$ and F iff $[\![A]\!]_g \neq [\![B]\!]_g$

7. $[\![\star]\!]_g = \text{N}$

Abbreviations

1. $\varphi \vee \psi$ abbreviates $\neg(\neg\varphi \wedge \neg\psi)$
2. $\varphi \leftrightarrow \psi$ abbreviates $\neg(\varphi \wedge \neg\psi)$
3. $\forall x \varphi$ abbreviates $\neg\exists x \neg\varphi$
4. $\varphi \dot{\wedge} \psi$ abbreviates $(\varphi \wedge \psi) \vee (\neg\varphi \wedge \varphi)$
5. $\varphi \ddot{\wedge} \psi$ abbreviates $(\varphi \wedge \psi) \vee (\neg\varphi \wedge \varphi) \vee (\neg\psi \wedge \psi)$
6. $\top$ abbreviates $\star \equiv \star$
7. $\partial\pi$ abbreviates $(\pi \equiv \top) \vee \star$
8. $\varphi_{\langle\pi\rangle}$ abbreviates $\partial\pi \dot{\wedge} \varphi$

Basic idea: Because presuppositions are built into the semantics, it is possible to provide a compositional account of presupposition and quantification that does not run into the binding problem of K&P.

2 Presuppositional Montague Grammar

We want to provide an analysis of sentences that contain both quantifiers and presuppositions:

(2) a. Somebody managed to succeed George V on the throne of England.
b. A fat man pushes his bicycle.
c. Every man who serves his king will be rewarded.
d. Every fat man pushes his bicycle.

To account for the data in (2), some additions must be made to the PTQ-fragment:

- Presupposition triggers:
  - *manage to* as a presuppositional variant of *try to*
  - *'s* as member of category *DET/NP*
- Adjectives:
  - *fat* as member of category *CN/CN*

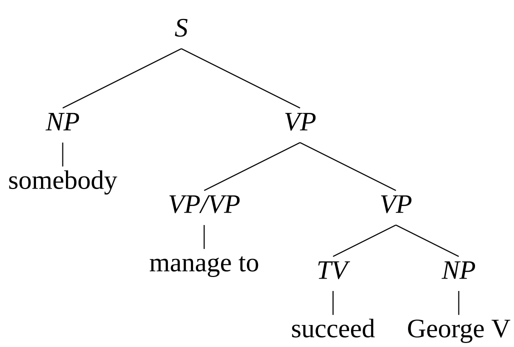
It is also necessary to define a function $(.)^\bullet$ that translates syntactic trees into expressions of TY_2[3].

- These expressions are then combined via functional application:

(3) $([[\alpha]^{A/^m B}[\beta]^B]^A)^\bullet = \alpha^\bullet\beta^\bullet$, for $m \in \{1, 2\}$

An example:

(4)



The translations of the relevant lexical entries into $\textsc{ty}_2$[3] are:

(5) a. somebody$^\bullet = \lambda P \lambda i \exists y(P\ yi)$
b. manage to$^\bullet = \lambda P \lambda x \lambda i(P\ xi_{\langle(\mathit{difficult}\ P)\ xi\rangle})$
c. succeed$^\bullet = \lambda Q \lambda y(Q \lambda x(\mathit{succeed}\ xy))$
d. George V$^\bullet = \lambda P(P\ g)$

By functional application:

1. ([succeed George V])$^\bullet$ = succeed$^\bullet$ George$^\bullet$
$\lambda Q \lambda y(Q \lambda x(\mathit{succeed}\ xy)) \lambda P(P\ g) \Rightarrow \lambda y(\mathit{succeed}\ gy) \Rightarrow \mathit{succeed}\ g$
2. ([manage to [succeed George V]])$^\bullet$ = manage to$^\bullet$ (succeed George V)$^\bullet$
$\lambda P \lambda x \lambda i(P\ xi_{\langle(\mathit{difficult} P)\ xi\rangle})\mathit{succeed}\ g \Rightarrow \lambda x \lambda i(\mathit{succeed}\ gxi_{\langle(\mathit{difficult}(\mathit{succeed}\ g))\ xi\rangle})$
3. ([somebody [manage to [succeed George V]]])$^\bullet$ = somebody$^\bullet$ (manage to succeed George V)$^\bullet$
$\lambda P \lambda i \exists y(P\ yi) \lambda x \lambda i(\mathit{succeed}\ gxi_{\langle(\mathit{difficult}(\mathit{succeed}\ g))\ xi\rangle}) \Rightarrow$
$\lambda P \lambda i \exists y(P\ yi) \lambda x \lambda j(\mathit{succeed}\ gxj_{\langle(\mathit{difficult}(\mathit{succeed}\ g))\ xj\rangle}) \Rightarrow \lambda i \exists y(\mathit{succeed}\ gyi_{\langle(\mathit{difficult}(\mathit{succeed}\ g))\ yi\rangle})$

Result: Function from states to truth values

- True if it is asserted that there is someone who succeeded George V in s and presupposed that that person found it difficult to succeed George V in s.
- Aviods K&P binding problem.

Defining presuppositions:

So far, we have a compositional account of presupposition that works if we stipulate the presuppositions that are associated with a particular proposition. It does not predict which presuppositions should be associated with which propositions.

In this intensional system, propositions are not simply True or False—they are True or False with respect to some state s.

- Must define when an expression φ_{st} presupposes an expression π_{st}:
- Let φ and π be expressions of type st. φ *presupposes* π iff for all models M, assignments g and for all states s:

(6) if $[\![\varphi s]\!]_{M,g} = $ T or $[\![\varphi s]\!]_{M,g} = $ F, then $[\![\pi s]\!]_{M,g} = $ T

- (Maximal) presupposition of φ: if φ is of the form $\lambda i \psi$ and ψ is a λ-free formula, then the presupposition of φ is given by:

(7) $\textsc{pr}(\varphi) = \lambda j(\textsc{tr}^+(\varphi j) \vee \textsc{tr}^-(\varphi j))$

Applied to our example, this definition produces:

(8) $\lambda j(\exists y((difficult(succeed\ g))\ yj \wedge succeed\ gyj) \vee \forall y((difficult(succeed\ g))\ yj \wedge \neg succeed\ gyj))$

Other presuppositions:

Every and *some* trigger existential presuppositions.

- In other words, the antecedent of material implication is always met.

In Presuppositional Montague Grammar, this presupposition means that *every* has the following translation:

(9) every$^\bullet = \lambda P_1 \lambda P_2 \lambda i(\forall x(P_1\ xi \dot{\rightarrow} P_2\ xi)_{\langle \exists y(P_1\ yi)\rangle})$

Some examples:

Both of the sentences in (10) presuppose that there was a (i.e., at least one) girl at the party:

(10) a. Bill kissed every girl at the party.
$\lambda i(\forall x(girl\ xi \dot{\rightarrow} kiss\ xbi)_{\langle \exists y(girl\ yi)\rangle})$

b. Bill didn't kiss every girl at the party.
$\lambda i\neg(\forall x(girl\ xi \dot{\rightarrow} kiss\ xbi)_{\langle \exists y(girl\ yi)\rangle})$

Stop doing X in a state s presupposes having done X before and asserts not doing X any more in s:

(11) stop to$^{\bullet} = \lambda P\lambda x\lambda i(\neg(P\ xi)_{\langle \exists j(j<i\ \dot{\wedge}\ i\approx j\ \dot{\wedge}\ P\ xj)\rangle})$

Applying this translation to an actual sentence gives us the translation in (12b):

(12) a. Somebody stopped dating Mary.

b. $\lambda i\exists x(\neg(date\ mxi)_{\langle \exists j(j<i\ \dot{\wedge}\ i\approx j\ \dot{\wedge}\ date\ mxj)\rangle})$

- Given a state s, it is asserted that there is someone who doesn't date Mary in s, and it is presupposed that there is a state s' which precedes s on the time-axis in which that *same* someone dates Mary.
- Presupposition: Either somebody has been dating Mary before and now no longer dates her, or everybody dated Mary before and still dates her at the moment.

The presupposition triggered by *too* requires a meaning postulate and a rule for a special kind of quantifying-in (cf. K&P):

(13) a. Too rule translated
$([[\xi]^{NP}[\vartheta]^{S}]_{\text{too}}{}^{S,n})^{\bullet} = \lambda i(\xi^{\bullet}\lambda x_n(\vartheta^{\bullet})i_{\langle too\xi^{\bullet}\vartheta^{\bullet}i\rangle})$

b. Meaning postulate
$too \equiv \lambda Q\lambda P\lambda i\exists x(*\ xi\ \dot{\wedge}\ \neg(\lambda P(P\ x) \equiv Q)\ \dot{\wedge}\ \lambda x_n(P)xi)$

The sentence in (14) asserts that Bill loves Mary and presupposes that there is someone other than Mary whom Bill loves as well:

(14) a. Bill loves Mary too.

b. $\lambda i(love\ mbi_{\langle \exists x(\neg(x\equiv m)\ \dot{\wedge}\ love\ xbi)\rangle})$

A problem:

Presuppositional Montague Grammar does not assign the correct presuppositions to texts like the following:

(15) a. A fat man pushes his bicycle. It is broken.

b. Every man who serves his king is rewarded by him.

This is not really a problem with presuppositions; it is a problem with classical Montague grammar and, indeed, with all static logics.

The solution? Dynamification!

3 Dynamifying Presuppositional Montague Grammar

To replace the system described in the previous section with a dynamic one, we need only four additions to $\textsc{ty}_2{}^3$:

- $DR_{(se)t}$, $\mapsto$, $i[d]j$, and three axioms

Step One:

- Add discourse referents $d_1, d_1, \ldots$ to $\textsc{ty}_2{}^3$.
  - Discourse markers are of type se (individual concepts).
- Add the non-logical constant DR of type $(se)t$
  - Interpretation: is a discourse referent.
  - The interpretation is total: every expression of type se either is a discourse referent or it is not.

  Add Muskens' classical implication $\mapsto$:

(16) $\varphi \mapsto \psi$ abbreviates $\varphi \wedge \psi \equiv \varphi$

Step Two:

- Define $i[d]j$ to mean: states i and j agree on all discourse referents except possibly in the value of d:

(17) $i[d]j$ iff $\forall d'((DR\, d' \;\wedge\; \neg(d \equiv d')) \mapsto d'i \equiv d'j)$

- Three axioms are required to make this work:

(18) a. AX1: $\forall i \forall d \forall x(DR\, d \mapsto \exists j(i[d]j \;\wedge\; dj = x))$
b. AX2: $DR\, d$, for each discourse referent d
c. AX3: $\neg(d_1 \equiv d_2)$, for each two different discourse referents d_1 and d_2

Step Three:

- Build the discourse fragment.
  - Sentences are translated into expressions of type $s(st)$ (relations between states).
- Add a rule for text formation:

(19) If $[\xi]^S$ and $[\vartheta]^S$ are trees, then $[[\xi]^S[\vartheta]^S]_{\mathrm{tf}}{}^S$ is a tree, and $([[\xi]^S[\vartheta]^S]_{\mathrm{tf}}{}^S)^\bullet = \xi^\bullet \,\dot{\wedge}\, \vartheta^\bullet$

Making the logic dynamic does not introduce any partiality into the system. In other words,modeling dynamics in $\textsc{ty}_2{}^3$ is exactly like modeling dynamics in $\textsc{ty}_2{}^2$.

This is not a dynamic semantics; rather, it is a mechanism for modeling dynamic discourse assignment in a static system.

The system is no longer intensional—states are assignments to discourse referents (not world-time pairs).

4 Conclusion

Benefits of (Dynamified) Presuppositional Montague Grammar:

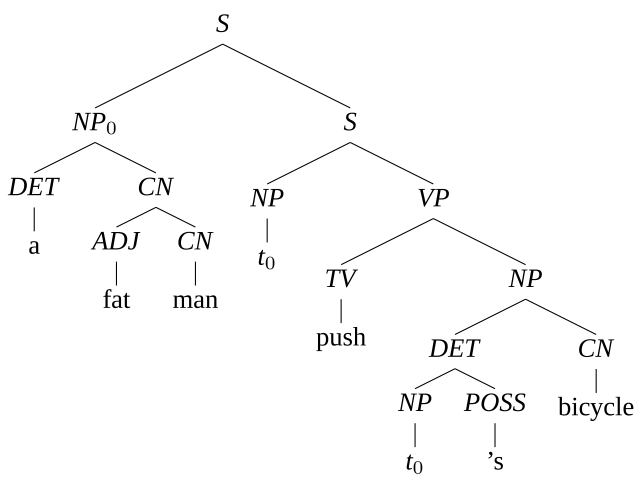
- Does not have K&P binding problem.
- Does not predict Heim's overly strong, universal presuppositions
- Makes the same predictions as Partial Predicate logic, but is fully compositional.
- Is compatible with K&P's fragment.

In this account, presupposition failure leads to undefinedness. One possible objection is that these "presuppositions" are really conventional implicatures; failing implicatures should not necessarily lead to undefined truth values.

- It is possible to turn $\text{TY}_2{}^3$ into $\text{TY}_2{}^4$ (a four-valued, two-sorted logic) with the addition of $\#$.
  - This is exactly the same system, except that $D_t = \{\mathrm{T}, \mathrm{F}, \mathrm{t}, \mathrm{f}\}$, where:
    - $[\![\star]\!] = \mathrm{t}$ (true in spite of presupposition failure)
    - $[\![\#]\!] = \mathrm{f}$ (false in spite of presupposition failure)

Appendix: More examples

(20)



This sentence requires a translation rule for quantifying-in:

(21) $([[\xi]^{NP}[\vartheta]^{S}]^{S,n}{}_{\mathrm{qi}})^{\bullet} = \xi^{\bullet}\lambda x_n(\vartheta^{\bullet})$, for $n \in \mathbb{N}$

The translation rules for the lexical items in the sentences are the following:

(22) a. $\text{a}^{\bullet} = \lambda P_1 \lambda P_2 \lambda i \exists y(P_1\ yi \mathbin{\dot\wedge} P_2\ yi)$
b. $\text{fat}^{\bullet} = fat$
c. $\text{man}^{\bullet} = man$
d. $\text{bicycle}^{\bullet} = bike$
e. $\text{push}^{\bullet} = \lambda Q \lambda y(Q \lambda x(push\ xy))$
f. $t_n{}^{\bullet} = \lambda P(P x_n)$
g. $\text{'s}^{\bullet} = \lambda Q \lambda P_1 \lambda P_2 \lambda i(\exists x(P_1\ xi \mathbin{\dot\wedge} Q\lambda y(of\ yx)i \mathbin{\dot\wedge} P_2\ xi)_{\langle \exists! x(P_1\ xi \mathbin{\dot\wedge} Q\lambda y(of\ yx)i)\rangle})$

Once we add a notation convention for the meaning of of, we can use functional application to get a compositional translation of the sentence:

(23) $\forall x \forall y \forall i((\gamma\ xi \mathbin{\dot\wedge} of\ xyi) \mathbin{\dot\rightarrow} \gamma - of\ yxi)$, where γ is man, $bike$ or $king$

1. ([fat man])$^\bullet \ \Rightarrow\ fat\ man$
2. ([a [fat man]])$^\bullet \ \Rightarrow\ \lambda P_3 \lambda i \exists z((fat\ man)\ zi \mathbin{\dot\wedge} P_3\ zi)$
3. ([t_0 's])$^\bullet \ \Rightarrow\ \lambda P_1 \lambda P_2 \lambda i(\exists x(P_1\ xi \mathbin{\dot\wedge} of\ x_0 x_i P_2\ xi)_{\langle \exists ! x(P_1\ xi\ \wedge\ of\ x_0 xi)\rangle})$
4. ([[t_0 's] bicycle])$^\bullet \ \Rightarrow\ \lambda P_2 \lambda i(\exists x(bike - of\ x_0 xi \mathbin{\dot\wedge} P_2\ xi)_{\langle \exists ! x(bike-of\ x_0 xi)\rangle})$
5. ([push [[t_0 's] bicycle]])$^\bullet \ \Rightarrow\ \lambda y \lambda i(\exists x(bike - of\ x_0 xi \mathbin{\dot\wedge} push\ xx_0 i)_{\langle \exists ! x(bike-of\ x_0 xi)\rangle})$
6. ([t_0 [push [[t_0 's] bicycle]]])$^\bullet \ \Rightarrow\ \lambda i(\exists x(bike - of\ x_0 xi \mathbin{\dot\wedge} push\ xx_0 i)_{\langle \exists ! x(bike-of\ x_0 xi)\rangle})$
7. ([[a [fat man]] t_0 [push [[t_0 's] bicycle]]]$_{\mathrm{qi}}{}^{S,0}$)$^\bullet \ \Rightarrow\ \lambda i \exists z((fat\ man))\ zi \mathbin{\dot\wedge} \exists x(bike{-}of\ zxi \mathbin{\dot\wedge} push\ xzi)_{\langle \exists ! x(bike-ofzxi)\rangle}$

The presupposition for this sentence is built up compositionally; it is the disjunction of an existential truth-condition and a universal falsity condition, which is weaker than the presupposition predicted in Heim (1983):

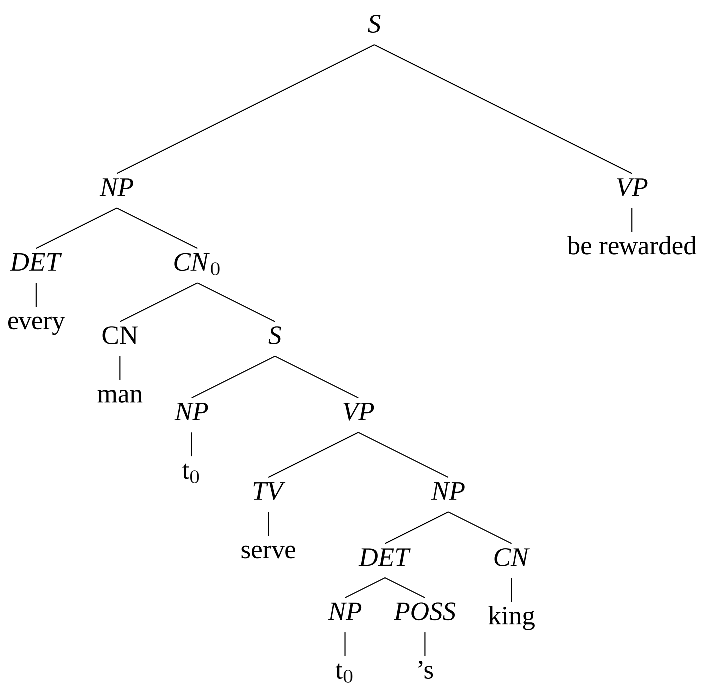
(24) $\lambda j(\exists z(fat\ man)\ zj\ \wedge\ \exists ! x(bike - of\ zxj)\ \wedge\ \exists x(bike - of\ zxj\ \wedge\ push\ xzj))\ \vee$
$\forall z((fat\ man)\ zj\ \rightarrow\ (\exists ! x(bike - of\ zxj)\ \wedge\ \forall x(bike - of\ zxj\ \rightarrow\ \neg push\ xzj)))$

The same procedure derives the following analogous sentence-translation pair:

(25) a. Every fat man pushes his bicycle.
b. $\lambda i \forall z((fat\ man)zi \mathbin{\dot\rightarrow} \exists x(bike - of\ zxi \mathbin{\dot\wedge} push\ xzi)_{\langle \exists ! x(bike-of\ zxi)\rangle})$

Sentences containing relative clauses require a rule of translation for relative clause formation:

(26) a.



b. $([[\xi]^{CN} [\vartheta]^{S}]_{\mathrm{rcf}}{}^{CN,n})^\bullet = \lambda x_n \lambda i(\xi^\bullet x_n i \mathbin{\dot\wedge} \vartheta^\bullet i)$, for $n \in \mathbb{N}$

c. ([[every [man [t_0 [serve [[t_0 's] king]]]]$_{\mathrm{rcf}}{}^{CN,0}$] be rewarded])$^\bullet \ \Rightarrow$
$\lambda i \forall x((man\ xi \mathbin{\dot\wedge} \exists y(king - of\ xyi \mathbin{\dot\wedge} serve\ yxi)_{\langle \exists ! yking-of\ xyi\rangle}) \mathbin{\dot\rightarrow} reward\ xi)$