Dynamic Montague Grammar Lite

Martin Jansche

November 1998

"... we are convinced that the capacities of [Montague Grammar] have not been exploited to the limit, that sometimes an analysis is carried out in a rival framework simply because it is more fashionable."

—— Groenendijk & Stokhof,
'Dynamic Montague Grammar'

Some definitions and some facts

Definition 1 (uparrow)
If neither p nor g occurs freely in ϕ:

$\uparrow : p \rightarrow d$
$\uparrow : (s \rightarrow t) \rightarrow ((s \rightarrow t) \rightarrow s \rightarrow t)$

$\uparrow\phi := \lambda p.\, \lambda g.\, (\phi(g) \wedge p(g))$

Definition 2 (downarrow)
$\downarrow : d \rightarrow p$
$\downarrow : ((s \rightarrow t) \rightarrow p) \rightarrow p$

$\downarrow\Phi := \Phi(\lambda g.\, \top)$

Definition 3 (truth)
True : $d \rightarrow t$
True : $(p \rightarrow s \rightarrow t) \rightarrow t$

True$(\Phi) := \forall_s(\downarrow\Phi)$

Fact 1 ($\downarrow\uparrow$-elimination) $\downarrow\uparrow\phi = \phi$

$$
\begin{array}{lll}
 & \downarrow(\uparrow\phi) & \\
= & \downarrow[\lambda p.\, \lambda g.\, (\phi(g) \wedge p(g))] & g \text{ not free in } \phi \\
= & [\lambda p.\, \lambda g.\, (\phi(g) \wedge p(g))](\lambda h.\, \top) & \\
\rightarrow_\beta & \lambda g.\, (\phi(g) \wedge [\lambda h.\, \top](g)) & \\
\rightarrow_\beta & \lambda g.\, (\phi(g) \wedge \top) & \\
= & \lambda g.\, \phi(g) & \\
\rightarrow_\eta & \phi & \text{since } g \text{ not free in } \phi
\end{array}
$$

Fact 2 (failure of ↑↓-elimination) $\uparrow\downarrow\Phi \neq \Phi$

$$
\begin{array}{ll}
 & \uparrow(\downarrow\Phi) \\
= & \uparrow(\Phi(\lambda h.\, \top)) \\
= & \lambda p.\, \lambda g.\, (\Phi(\lambda h.\, \top)(g) \wedge p(g)) \\
 & \text{Let } \Phi := \lambda p'.\, \lambda g'.\, (\phi(g') \wedge p'(k)) \text{, then continue:} \\
= & \lambda p.\, \lambda g.\, ([\lambda p'.\, \lambda g'.\, (\phi(g') \wedge p'(k))](\lambda h.\, \top)(g) \wedge p(g)) \\
\rightarrow_\beta & \lambda p.\, \lambda g.\, ([\lambda g'.\, (\phi(g') \wedge [\lambda h.\, \top](k))](g) \wedge p(g)) \\
\rightarrow_\beta & \lambda p.\, \lambda g.\, ((\phi(g) \wedge [\lambda h.\, \top](k)) \wedge p(g)) \\
\rightarrow_\beta & \lambda p.\, \lambda g.\, ((\phi(g) \wedge \top) \wedge p(g)) \\
= & \lambda p.\, \lambda g.\, (\phi(g) \wedge p(g)) =_\alpha \lambda p'.\, \lambda g'.\, (\phi(g') \wedge p'(g'))
\end{array}
$$

Definition 4 (static negation)
If g has no free occurrence in Φ:

$$\sim\, : d \rightarrow d$$
$$\sim\, : (p \rightarrow s \rightarrow t) \rightarrow d$$
$$\sim\Phi := \uparrow[\lambda g.\, \neg((\downarrow\Phi)(g))]$$

N.B.: If $\neg$ is generalized negation, define $\sim\Phi := \uparrow\neg\downarrow\Phi$.

Fact 3 $\sim\sim\Phi = \uparrow\downarrow\Phi$

$$
\begin{array}{lll}
\sim\sim\Phi & = & \uparrow[\lambda g.\, \neg(\downarrow\sim\Phi)(g)] \\
 & = & \uparrow[\lambda g.\, \neg(\downarrow\uparrow[\lambda g'.\, \neg(\downarrow\Phi)(g')])(g)] \\
 & = & \uparrow[\lambda g.\, \neg[\lambda g'.\, \neg(\downarrow\Phi)(g')](g)] \\
 & \rightarrow_\beta & \uparrow[\lambda g.\, \neg\neg(\downarrow\Phi)(g)] \\
 & = & \uparrow[\lambda g.\, (\downarrow\Phi)(g)] \\
 & \rightarrow_\eta & \uparrow(\downarrow\Phi)
\end{array}
$$

Definition 5 (dynamic conjunction)
If p has no free occurrence in Φ or Ψ:

$$⨟ : d \to d \to d$$
$$⨟ : (p \to p) \to (p \to p) \to p \to p$$
$$(\Phi \mathbin{⨟} \Psi) := \lambda p.\, \Phi(\Psi(p))$$

Definition 6 (update)
What it means to update an assignment function:

$$\text{update} : m \to e \to s \to s$$
$$\text{update} : m \to e \to (m \to e) \to m \to e$$
$$\text{update}(d)(x)(g)(d) := x$$
$$\text{update}(d)(x)(g)(d') := g(d') \text{ provided } d \neq d'$$

Generously add syntactic sugar: $\{x/d\}g :=$ update$(d)(x)(g)$

Definition 7 (dynamic existential quantifier)
If there are no free occurrences of p, g, x in Φ:

$$\mathcal{E} : m \to d \to d$$
$$\mathcal{E} : m \to (p \to s \to t) \to p \to s \to t$$
$$\mathcal{E} d \Phi := \lambda p.\, \lambda g.\, \exists x \Phi(p)(\{x/d\}g)$$

Definition 8 (remaining connectives)
1. internally dynamic implication: $(\Phi \Rightarrow \Psi) := \sim(\Phi \mathbin{⨟} \sim\Psi)$
2. static disjunction: $(\Phi \ \mathbf{or}\ \Psi) := (\sim\Phi \Rightarrow \Psi)$
3. static universal quanitifier: $\mathcal{A} d \Phi := \sim\mathcal{E} d \sim\Phi$

Fact 4 $\sim\mathcal{E}d\Phi = \mathcal{A}d{\sim}\Phi$

$$
\begin{array}{ll}
 & \sim\mathcal{E}d\Phi \\
= & \uparrow[\lambda g.\, \neg(\downarrow\mathcal{E}d\Phi)(g)] \\
= & \uparrow[\lambda g.\, \neg(\downarrow[\lambda p'.\, \lambda g'.\, \exists x\Phi(p')(\{x/d\}g')])(g)] \\
= & \uparrow[\lambda g.\, \neg[\lambda p'.\, \lambda g'.\, \exists x\Phi(p')(\{x/d\}g')](\lambda h.\, \top)(g)] \\
\rightarrow_\beta & \uparrow[\lambda g.\, \neg[\lambda g'.\, \exists x\Phi(\lambda h.\, \top)(\{x/d\}g')](g)] \\
\rightarrow_\beta & \uparrow[\lambda g.\, \neg\exists x\Phi(\lambda h.\, \top)(\{x/d\}g)]
\end{array}
$$

$$
\begin{array}{ll}
 & \mathcal{A}d{\sim}\Phi \\
= & \sim\mathcal{E}d{\sim}{\sim}\Phi \\
= & \sim\mathcal{E}d{\uparrow}{\downarrow}\Phi \\
= & \uparrow[\lambda g.\, \neg\exists x[\uparrow\downarrow\Phi](\lambda h.\, \top)(\{x/d\}g)] \\
= & \uparrow[\lambda g.\, \neg\exists x[\lambda p'.\, \lambda g'.\, ((\downarrow\Phi)(g') \wedge p'(g'))](\lambda h.\, \top)(\{x/d\}g)] \\
\rightarrow_\beta & \uparrow[\lambda g.\, \neg\exists x[\lambda g'.\, ((\downarrow\Phi)(g') \wedge [\lambda h.\, \top](g'))](\{x/d\}g)] \\
= & \uparrow[\lambda g.\, \neg\exists x[\lambda g'.\, ((\downarrow\Phi)(g') \wedge \top)](\{x/d\}g)] \\
= & \uparrow[\lambda g.\, \neg\exists x[\lambda g'.\, (\downarrow\Phi)(g')](\{x/d\}g)] \\
\rightarrow_\beta & \uparrow[\lambda g.\, \neg\exists x(\downarrow\Phi)(\{x/d\}g)] \\
= & \uparrow[\lambda g.\, \neg\exists x\Phi(\lambda h.\, \top)(\{x/d\}g)]
\end{array}
$$

Fact 5 $\mathcal{E}d\Phi \,;\, \Psi = \mathcal{E}d(\Phi \,;\, \Psi)$

$$
\begin{array}{ll}
 & \mathcal{E}d(\Phi \,;\, \Psi) \\
= & \lambda p.\, \lambda g.\, \exists x(\Phi \,;\, \Psi)(p)(\{x/d\}g) \\
= & \lambda p.\, \lambda g.\, \exists x[\lambda p'.\, \Phi(\Psi(p'))](p)(\{x/d\}g) \\
\longrightarrow_\beta & \lambda p.\, \lambda g.\, \exists x\Phi(\Psi(p))(\{x/d\}g)
\end{array}
$$

$$
\begin{array}{ll}
 & \mathcal{E}d\Phi \,;\, \Psi \\
= & \lambda p.\, [\mathcal{E}d\Phi](\Psi(p)) \\
= & \lambda p.\, [\lambda p'.\, \lambda g.\, \exists x\Phi(p')(\{x/d\}g)](\Psi(p)) \\
\longrightarrow_\beta & \lambda p.\, \lambda g.\, \exists x\Phi(\Psi(p))(\{x/d\}g)
\end{array}
$$

Fact 6 $(\mathcal{E}d\Phi \Rightarrow \Psi) = \mathcal{A}d(\Phi \Rightarrow \Psi)$

$$
\begin{array}{ll}
(\mathcal{E}d\Phi \Rightarrow \Psi) & \\
= \sim(\mathcal{E}d\Phi \,;\, \sim\Psi) & \text{definition of } \Rightarrow \\
= \sim\mathcal{E}d(\Phi \,;\, \sim\Psi) & \text{Fact 5} \\
= \mathcal{A}d\sim(\Phi \,;\, \sim\Psi) & \text{Fact 4} \\
= \mathcal{A}d(\Phi \Rightarrow \Psi) & \text{definition of } \Rightarrow
\end{array}
$$

Dynamic Montague Grammar

Say $P(\!|d|\!) := \lambda g.\, P(g(d))$ for all $P : e \rightarrow t$.

Definition 9 (translation of basic expressions)

$$
\begin{array}{lll}
[\![a^i]\!] & := \lambda P.\, \lambda Q.\, \mathcal{E}\mathsf{d}_i(P(\mathsf{d}_i) \,;\, Q(\mathsf{d}_i)) & : (m \rightarrow d) \rightarrow (m \rightarrow d) \rightarrow d \\
[\![man]\!] & := \lambda d.\, \uparrow\mathsf{man}(\!|d|\!) & : m \rightarrow d \\
 & = \lambda d.\, \lambda p.\, \lambda g.\, (\mathsf{man}(g(d)) \wedge p(g)) & \\
[\![walks]\!] & := \lambda d.\, \uparrow\mathsf{walk}(\!|d|\!) & : m \rightarrow d \\
[\![he_i]\!] & := \lambda Q.\, Q(\mathsf{d}_i) & : (m \rightarrow d) \rightarrow d \\
[\![talks]\!] & := \lambda d.\, \uparrow\mathsf{talk}(\!|d|\!) & : m \rightarrow d
\end{array}
$$

$[\![a^1\ man]\!] = [\![a^1]\!]([\![man]\!])$

$= [\lambda P.\, \lambda Q.\, \mathcal{E}\mathsf{d}_1(P(\mathsf{d}_1) \,;\, Q(\mathsf{d}_1))]([\![man]\!])$

$= \lambda Q.\, \mathcal{E}\mathsf{d}_1([\![man]\!](\mathsf{d}_1) \,;\, Q(\mathsf{d}_1))$

$= \lambda Q.\, \mathcal{E}\mathsf{d}_1([\lambda d.\, \lambda p.\, \lambda g.\, (\mathsf{man}(g(d)) \wedge p(g))](\mathsf{d}_1) \,;\, Q(\mathsf{d}_1))$

$= \lambda Q.\, \mathcal{E}\mathsf{d}_1([\lambda p.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge p(g))] \,;\, Q(\mathsf{d}_1))$

$$\llbracket a^1 \ man \ walks \rrbracket = \llbracket a^1 \ man \rrbracket (\llbracket walks \rrbracket)$$

$$= [\lambda Q.\, \mathcal{E}\mathsf{d}_1([\lambda p.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge p(g))] \mathbin{;} Q(\mathsf{d}_1))](\llbracket walks \rrbracket)$$

$$= \mathcal{E}\mathsf{d}_1([\lambda p.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge p(g))] \mathbin{;} \llbracket walks \rrbracket(\mathsf{d}_1))$$

$$= \mathcal{E}\mathsf{d}_1([\lambda p.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge p(g))] \mathbin{;} [\lambda d.\, \lambda p'.\, \lambda g'.\, (\mathsf{walk}(g'(d)) \wedge p'(g'))](\mathsf{d}_1))$$

$$= \mathcal{E}\mathsf{d}_1([\lambda p.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge p(g))] \mathbin{;} [\lambda p'.\, \lambda g'.\, (\mathsf{walk}(g'(\mathsf{d}_1)) \wedge p'(g'))])$$

$$= \mathcal{E}\mathsf{d}_1 \lambda p''.\, [\lambda p.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge p(g))]([\lambda p'.\, \lambda g'.\, (\mathsf{walk}(g'(\mathsf{d}_1)) \wedge p'(g'))](p''))$$

$$= \mathcal{E}\mathsf{d}_1 \lambda p''.\, [\lambda p.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge p(g))](\lambda g'.\, (\mathsf{walk}(g'(\mathsf{d}_1)) \wedge p''(g')))$$

$$= \mathcal{E}\mathsf{d}_1 \lambda p''.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge [\lambda g'.\, (\mathsf{walk}(g'(\mathsf{d}_1)) \wedge p''(g'))](g))$$

$$= \mathcal{E}\mathsf{d}_1 \lambda p''.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge \mathsf{walk}(g(\mathsf{d}_1)) \wedge p''(g))$$

$$= \lambda p'.\, \lambda g'.\, \exists x [\lambda p''.\, \lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge \mathsf{walk}(g(\mathsf{d}_1)) \wedge p''(g))](p')(\{x/\mathsf{d}_1\}g')$$

$$= \lambda p'.\, \lambda g'.\, \exists x [\lambda g.\, (\mathsf{man}(g(\mathsf{d}_1)) \wedge \mathsf{walk}(g(\mathsf{d}_1)) \wedge p'(g))](\{x/\mathsf{d}_1\}g')$$

$$= \lambda p'.\, \lambda g'.\, \exists x (\mathsf{man}(\{x/\mathsf{d}_1\}g'(\mathsf{d}_1)) \wedge \mathsf{walk}(\{x/\mathsf{d}_1\}g'(\mathsf{d}_1)) \wedge p'(\{x/\mathsf{d}_1\}g'))$$

$$= \lambda p'.\, \lambda g'.\, \exists x (\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge p'(\{x/\mathsf{d}_1\}g'))$$

$$[\![he_1\ talks]\!] = [\![he_1]\!]([\![talks]\!])$$

$$= [\lambda Q.\, Q(\mathsf{d}_1)]([\![talks]\!])$$

$$= [\![talks]\!](\mathsf{d}_1)$$

$$= [\lambda d.\, \lambda p.\, \lambda g.\, (\mathsf{talk}(g(d)) \wedge p(g))](\mathsf{d}_1)$$

$$= \lambda p.\, \lambda g.\, (\mathsf{talk}(g(\mathsf{d}_1)) \wedge p(g))$$

$$[\![a^1\ man\ walks.\ he_1\ talks]\!] = [\![a^1\ man\ walks]\!] \,⨟\, [\![he_1\ talks]\!]$$

$$= \lambda p.\, [\![a^1\ man\ walks]\!]([\![he_1\ talks]\!](p))$$

$$= \lambda p.\, [\![a^1\ man\ walks]\!]([\lambda p'.\, \lambda g'.\, (\mathsf{talk}(g'(\mathsf{d}_1)) \wedge p'(g'))](p))$$

$$= \lambda p.\, [\![a^1\ man\ walks]\!](\lambda g'.\, (\mathsf{talk}(g'(\mathsf{d}_1)) \wedge p(g')))$$

$$= \lambda p.\, [\lambda p'.\, \lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge p'(\{x/\mathsf{d}_1\}g))](\lambda g'.\, (\mathsf{talk}(g'(\mathsf{d}_1)) \wedge p(g')))$$

$$= \lambda p.\, \lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge [\lambda g'.\, (\mathsf{talk}(g'(\mathsf{d}_1)) \wedge p(g'))](\{x/\mathsf{d}_1\}g))$$

$$= \lambda p.\, \lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(\{x/\mathsf{d}_1\}g(\mathsf{d}_1)) \wedge p(\{x/\mathsf{d}_1\}g))$$

$$= \lambda p.\, \lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x) \wedge p(\{x/\mathsf{d}_1\}g))$$

$\mathsf{True}([\![a^1 \ man\ walks.\ he_1\ talks]\!])$

$= \forall_s(\downarrow[\![a^1 \ man\ walks.\ he_1\ talks]\!])$

$= \forall_s(\downarrow[\lambda p.\, \lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x) \wedge p(\{x/\mathsf{d}_1\}g))])$

$= \forall_s([\lambda p.\, \lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x) \wedge p(\{x/\mathsf{d}_1\}g))](\lambda h.\, \top))$

$= \forall_s[\lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x) \wedge [\lambda h.\, \top](\{x/\mathsf{d}_1\}g))]$

$= \forall_s[\lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x) \wedge \top)]$

$= \forall_s[\lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x))]$

$= [\lambda g.\, \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x))] \equiv [\lambda g.\, \top]$

$= \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x)) \equiv \top$

$= \exists x(\mathsf{man}(x) \wedge \mathsf{walk}(x) \wedge \mathsf{talk}(x))$