Mathematical Logic and Linguistics

Glyn Morrill & Oriol Valentín

Department of Computer Science
Universitat Politècnica de Catalunya
morrill@cs.upc.edu & oriol.valentin@gmail.com

BGSMath Course
Class 10

Syntactic and Semantic Analyses: the PTQ Fragment

Syntactic and Semantic Analyses: the PTQ Fragment

”The Montague Test” is the challenge of providing a computational cover grammar of the Montague fragment.

Montague Grammar

Montague Grammar

The Montague fragment:

Montague Grammar

The Montague fragment:

- overturned the view that semantics cannot be formalised

Montague Grammar

The Montague fragment:

- overturned the view that semantics cannot be formalised
- generated a paradigm of logical semantics

Montague Grammar

The Montague fragment:

- overturned the view that semantics cannot be formalised
- generated a paradigm of logical semantics
- gave rise to a field of formal semantics (*L&P*, A'dam Colloquium, and for forth)

Montague Grammar

The Montague fragment:

- overturned the view that semantics cannot be formalised
- generated a paradigm of logical semantics
- gave rise to a field of formal semantics (*L&P*, A'dam Colloquium, and for forth)

The fragment includes:

Montague Grammar

The Montague fragment:

- overturned the view that semantics cannot be formalised
- generated a paradigm of logical semantics
- gave rise to a field of formal semantics (*L&P*, A'dam Colloquium, and for forth)

The fragment includes:

- quantification

Montague Grammar

The Montague fragment:

- overturned the view that semantics cannot be formalised
- generated a paradigm of logical semantics
- gave rise to a field of formal semantics (*L&P*, A'dam Colloquium, and for forth)

The fragment includes:

- quantification
- anaphora

Montague Grammar

The Montague fragment:

- overturned the view that semantics cannot be formalised
- generated a paradigm of logical semantics
- gave rise to a field of formal semantics (*L&P*, A'dam Colloquium, and for forth)

The fragment includes:

- quantification
- anaphora
- intensionality

Montague Grammar

The Montague fragment:

- overturned the view that semantics cannot be formalised
- generated a paradigm of logical semantics
- gave rise to a field of formal semantics (*L&P*, A'dam Colloquium, and for forth)

The fragment includes:

- quantification
- anaphora
- intensionality
- coordination

CatLog2

CatLog2

CatLog2 is a type logical parser/theorem prover. It:

CatLog2

CatLog2 is a type logical parser/theorem prover. It:

- comprises 6000 lines of prolog

CatLog2

CatLog2 is a type logical parser/theorem prover. It:

- comprises 6000 lines of prolog
- has 20 primitive categorial connectives, 29 defined connectives, and 1 metalogical connective: a total of 50

CatLog2

CatLog2 is a type logical parser/theorem prover. It:

- comprises 6000 lines of prolog
- has 20 primitive categorial connectives, 29 defined connectives, and 1 metalogical connective: a total of 50
- has typically 2 rules for each connective: a rule of use and a rule of proof: roughly $50 \times 2 = 100$ rules

CatLog2

CatLog2 is a type logical parser/theorem prover. It:

- comprises 6000 lines of prolog
- has 20 primitive categorial connectives, 29 defined connectives, and 1 metalogical connective: a total of 50
- has typically 2 rules for each connective: a rule of use and a rule of proof: roughly $50 \times 2 = 100$ rules
- uses backward chaining focused sequent proof search so that for a binary connective for half of the rules there are 4 cases: $+/+, +/-, -/+, -/-$: $50 \times 4 + 50 =$ a total of about 250 focused rules

The Montague Test

The Montague Test

Concretely, we define the Montague test as the task of:

The Montage Test

Concretely, we define the Montague test as the task of:

semantically parsing the mini-corpus comprising the example sentences of Chapter Seven of Dowty et al. (1981)

The Montage Test

Concretely, we define the Montague test as the task of:

semantically parsing the mini-corpus comprising the example sentences of Chapter Seven of Dowty et al. (1981)

as a baseline criterion for grammar formalisms.

The Montague Test

Concretely, we define the Montague test as the task of:

semantically parsing the mini-corpus comprising the example sentences of Chapter Seven of Dowty et al. (1981)

as a baseline criterion for grammar formalisms.

I.e. we propose as a sine qua non being able to simulate computationally the Montague syntax-semantics interface of quantification, anaphora, intensionality, coordination, ...

Taking on the Montague Test

We invoke CatLog2 on this mini-corpus . . .

```
str(dwp('(7-7)'), [b([john]), walks], s(f)).
str(dwp('(7-16)'), [b([every, man]), talks], s(f)).
str(dwp('(7-19)'), [b([the, fish]), walks], s(f)).
str(dwp('(7-32)'), [b([every, man]), b([b([walks, or, talks])])], s(f)).
str(dwp('(7-34)'), [b([b([b([every, man]), walks, or, b([every, man]), talks])])], s(f)).
str(dwp('(7-39)'), [b([b([b([a, woman]), walks, and, b([she]), talks])])], s(f)).
str(dwp('(7-43, 45)'), [b([john]), believes, that, b([a, fish]), walks], s(f)).
str(dwp('(7-48, 49, 52)'), [b([every, man]), believes, that, b([a, fish]), walks], s(f)).
str(dwp('(7-57)'), [b([every, fish, such, that, b([it]), walks]), talks], s(f)).
str(dwp('(7-60, 62)'), [b([john]), seeks, a, unicorn], s(f)).
str(dwp('(7-73)'), [b([john]), is, bill], s(f)).
str(dwp('(7-76)'), [b([john]), is, a, man], s(f)).
str(dwp('(7-83)'), [necessarily, b([john]), walks], s(f)).
str(dwp('(7-86)'), [b([john]), walks, slowly], s(f)).
str(dwp('(7-91)'), [b([john]), tries, to, walk], s(f)).
str(dwp('(7-94)'), [b([john]), tries, to, b([b([catch, a, fish, and, eat, it])])], s(f)).
str(dwp('(7-98)'), [b([john]), finds, a, unicorn], s(f)).
str(dwp('(7-105)'), [b([every, man, such, that, b([he]), loves, a, woman]), loses, her],
s(f)).
str(dwp('(7-110)'), [b([john]), walks, in, a, park], s(f)).

str(dwp('(7-116, 118)'), [b([every, man]), doesnt, walk], s(f)).
```

Categorial connectives

	cont. mult.	disc. mult.	add.	qu.	norm. mod.	brack. mod.	exp.	limited contr. & weak.
primary	/ \ • I	↑ ↓ ⊙ J	& ⊕	⋀ ⋁	□ ◇	$[]^{-1}$ ⟨⟩	! ?	| W
sem. inactive variants	•– –∘ ∘– –• ◐ ◑	⫯• ⫰∘ ⫯∘ ⫰• ◓ ◒	⊓ ⊔	∀ ∃	■ ♦			
det. synth.	$\triangleleft^{-1}$ $\triangleright^{-1}$ $\triangleleft$ $\triangleright$	ˇ ˆ						diff.
nondet. synth.	÷ ×	⇑ ⇓ ⊚						−

a: $\blacksquare\forall g(\forall f((Sf\uparrow\blacksquare Nt(s(g)))\downarrow Sf)/CNs(g)) : \lambda A\lambda B\exists C[(A\ C)\wedge(B\ C)]$

and: $\blacksquare\forall f((?\blacksquare Sf\backslash[\,]^{-1}[\,]^{-1}Sf)/\blacksquare Sf) : (\Phi^{n+}\ 0\ and)$

and: $\blacksquare\forall a\forall f((?\blacksquare(\langle\rangle Na\backslash Sf)\backslash[\,]^{-1}[\,]^{-1}(\langle\rangle Na\backslash Sf))/\blacksquare(\langle\rangle Na\backslash Sf)) : (\Phi^{n+}\ (s\ 0)\ and)$

believes: $\Box((\langle\rangle\exists gNt(s(g))\backslash Sf)/(CPthat\sqcup\Box Sf)) : {}^\wedge\lambda A\lambda B(Pres\ (({}^\vee believe\ A)\ B))$

bill: $\blacksquare Nt(s(m)) : b$

catch: $\Box((\langle\rangle\exists aNa\backslash Sb)/\exists aNa) : {}^\wedge\lambda A\lambda B(({}^\vee catch\ A)\ B)$

doesnt: $\blacksquare\forall g\forall a((Sg\uparrow((\langle\rangle Na\backslash Sf)/(\langle\rangle Na\backslash Sb)))\downarrow Sg) : \lambda A\neg(A\ \lambda B\lambda C(B\ C))$

eat: $\Box((\langle\rangle\exists aNa\backslash Sb)/\exists aNa) : {}^\wedge\lambda A\lambda B(({}^\vee eat\ A)\ B)$

every: $\blacksquare\forall g(\forall f((Sf\uparrow Nt(s(g)))\downarrow Sf)/CNs(g)) : \lambda A\lambda B\forall C[(A\ C)\rightarrow(B\ C)]$

finds: $\Box((\langle\rangle\exists gNt(s(g))\backslash Sf)/\exists aNa) : {}^\wedge\lambda A\lambda B(Pres\ (({}^\vee find\ A)\ B))$

fish: $\Box CNs(n) : fish$

he: $\blacksquare[\,]^{-1}\forall g((\blacksquare Sg|\blacksquare Nt(s(m)))/(\langle\rangle Nt(s(m))\backslash Sg)) : \lambda AA$

her: $\blacksquare\forall g\forall a(((\langle\rangle Na\backslash Sg)\uparrow\blacksquare Nt(s(f)))\downarrow(\blacksquare(\langle\rangle Na\backslash Sg)|\blacksquare Nt(s(f)))) : \lambda AA$

in: $\Box(\forall a\forall f((\langle\rangle Na\backslash Sf)\backslash(\langle\rangle Na\backslash Sf))/\exists aNa) : {}^\wedge\lambda A\lambda B\lambda C(({}^\vee in\ A)\ (B\ C))$

is: $\blacksquare((\langle\rangle\exists gNt(s(g))\backslash Sf)/(\exists aNa\oplus(\exists g((CNg/CNg)\sqcup(CNg\backslash CNg))-I))) : \lambda A\lambda B(Pres\ (A\rightarrow C.[B=C];D.((D\ \lambda E[E=B])\ B)))$

john: $\blacksquare Nt(s(m)) : j$

loses: $\Box((\langle\rangle\exists gNt(s(g))\backslash Sf)/\exists aNa) : {}^\wedge\lambda A\lambda B(Pres\ (({}^\vee lose\ A)\ B))$

loves: $\Box((\langle\rangle\exists gNt(s(g))\backslash Sf)/\exists aNa) : {}^\wedge\lambda A\lambda B(Pres\ (({}^\vee love\ A)\ B))$

man: $\Box CNs(m) : man$

necessarily: $\blacksquare(SA/\Box SA) : Nec$

or: $\blacksquare\forall f((?\blacksquare(Sf/(\langle\rangle\exists gNt(s(g))\backslash Sf))\backslash[\,]^{-1}[\,]^{-1}(Sf/(\langle\rangle\exists gNt(s(g))\backslash Sf)))/\blacksquare(Sf/(\langle\rangle\exists gNt(s(g))\backslash Sf))) : (\Phi^{n+}\ (s\ 0)\ or)$

park: $\Box CNs(n) : park$

seeks: $\Box((\langle\rangle\exists gNt(s(g))\backslash Sf)/\Box\forall a\forall f(((Na\backslash Sf)/\exists bNb)\backslash(Na\backslash Sf))) : {}^\wedge\lambda A\lambda B(({}^\vee tries\ {}^\wedge(({}^\vee A\ {}^\vee find)\ B))\ B)$

slowly: $\Box\forall a\forall f(\Box(\langle\rangle Na\backslash Sf)\backslash(\langle\rangle\Box Na\backslash Sf)) : {}^\wedge\lambda A\lambda B({}^\vee slowly\ {}^\wedge({}^\vee A\ {}^\vee B))$

such+that: $\blacksquare\forall n((CNn\backslash CNn)/(Sf|\blacksquare Nt(n))) : \lambda A\lambda B\lambda C[(B\ C)\wedge(A\ C)]$

talks: $\Box(\langle\rangle\exists gNt(s(g))\backslash Sf) : {}^\wedge\lambda A(Pres\ ({}^\vee talk\ A))$

that: $\blacksquare(CPthat/\Box Sf) : \lambda AA$

the: $\blacksquare\forall n(Nt(n)/CNn) : \iota$

to: $\blacksquare((PPto/\exists aNa)\sqcap\forall n((\langle\rangle Nn\backslash Si)/(\langle\rangle Nn\backslash Sb))) : \lambda AA$

unicorn: $\Box CNs(n) : unicorn$

walks: $\Box(\langle\rangle\exists gNt(s(g))\backslash Sf) : {}^\wedge\lambda A(Pres\ ({}^\vee walk\ A))$

woman: $\Box CNs(f) : woman$