Computational Semantics

Questions and gap threading

Questions

Any respectable natural language understanding system has to be able to process questions.

- Who did the male suspect meet?
  $?x(person(x), the_y(suspect(y) \wedge male(y), meet(y, x)))$
- What did the female suspect buy?
  $?x(location(x), the_y(suspect(y) \wedge female(y), go(y, x)))$
- Was there an explosion in the peer?
  $\exists x(explosion(x) \wedge the_y(peer(y) \wedge in(x, y)))$

$?_x(\phi) =$ the $g(x)$ that makes $\exists x(\phi)$ true

Questions

Yes-No Questions (polar questions)

(1) a. Did the flight depart?

b. Will the letter mention Robin?

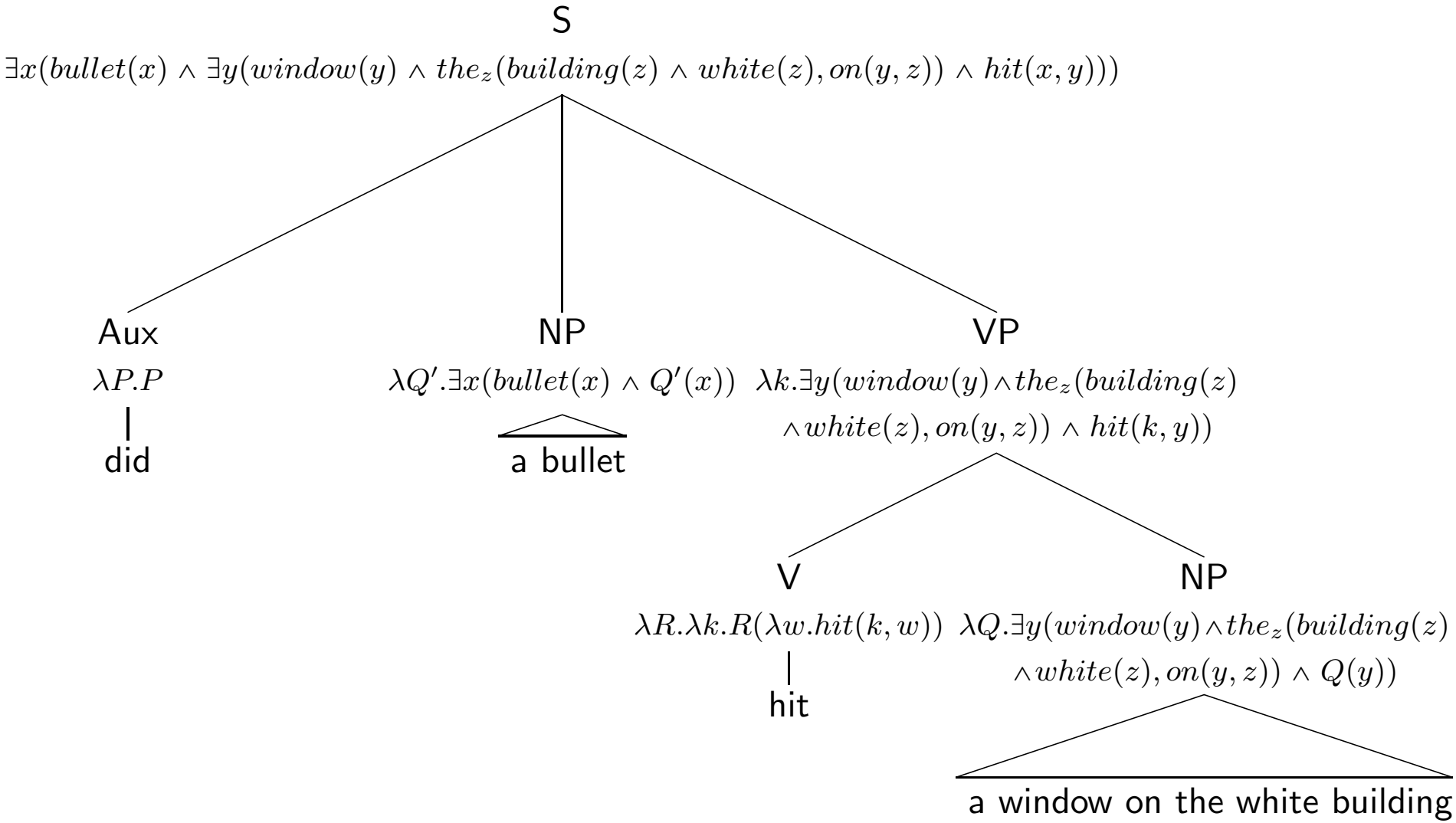
General syntactic-semantic pattern:

(S ; $(\theta)((\phi)(\psi))) \rightarrow$ (Aux ; θ) (NP; ϕ) (VP; ψ)
(Aux; $\lambda P.(\lambda e.future(e) \wedge P(e)) \rightarrow$ will
(Aux; $\lambda P.(\lambda e.past(e) \wedge P(e)) \rightarrow$ did

Let's simplify things and ignore tense: (Aux ; $\lambda P.P$) $\rightarrow$ will | did | ...

Questions

(2) Did a bullet hit a window on the white building?



Questions

The verb 'to be' is trickier

Predicative AP complements:

(3) a. Is Robin single?

$(\text{Adj}_{pred}, \lambda y.single(x) \wedge x = y) \rightarrow$ single

The above predicative adjective licenses (3a):

$$(\lambda Q.Q(robin))(\lambda y.single(x) \wedge x = y) =$$
$$(\lambda y.single(x) \wedge x = y)(robin) =$$
$$single(x) \wedge x = robin$$

b. Is the boy single?

$$(\lambda Q.the_k(boy(k) \wedge Q(k)))(\lambda y.single(x) \wedge x = y) =$$
$$the_k(boy(k) \wedge (\lambda y.single(x) \wedge x = y)(k)) =$$
$$the_k(boy(k) \wedge single(x) \wedge x = k)$$

Questions

Predicative NP complements:

(4) a. Is Robin a criminal?
$(\lambda Q.Q(robin))(\lambda y.criminal(x) \wedge x = y) =$
$criminal(x) \wedge x = robin$

b. Is the man a criminal?
$(\lambda Q.the_k(boy(k) \wedge Q(k)))(\lambda y.criminal(x) \wedge x = y) =$
$the_k(man(k) \wedge criminal(x) \wedge x = k)$

c. Is Robin at the house?
$(\lambda Q.Q(robin))(\lambda y.\exists x(house(x) \wedge at(y, x) \wedge x = y) =$
$\exists x(house(x) \wedge at(y, x) \wedge x = robin)$

We have therefore several patterns:

S → BE NP AP
S → BE NP PP
S → BE NP NP
BE → is | was | are

Questions

Wh-questions are the most interesting cases:

- **Subject interrogatives:**

  (5) a. Who died?
  b. Which suspect punched the guard?
  c. Who sent this letter to Robin?

  (6) a. Who do you think _ won the game?
  b. Which suspect was it that _ punched the guard?
  c. Who did you say _ sent this letter to Robin?

- **Complement interrogatives**

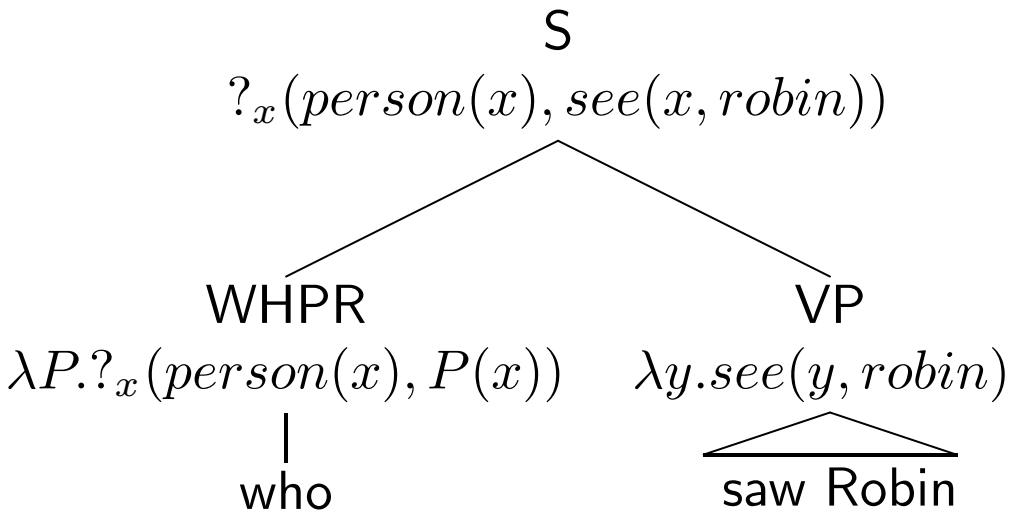
  (7) a. Who did Robin hug _ ?
  b. What did you say that Robin bought _ ?
  c. Which book did you say Kim wanted us to read _ ?

Questions

Non-embedded subject interrogatives are easy

(S; $(\phi)(\psi)$) $\rightarrow$ (WHPR; ϕ) (VP; ψ)

(WHPR; $\lambda P.?_x(person(x), P(x))$) $\rightarrow$ who
(WHPR; $\lambda P.?_x(thing(x), P(x))$) $\rightarrow$ what



$?_x(\phi)$ means 'provide the value of $g(x)$ such that $\exists x(\phi)$ is true'.

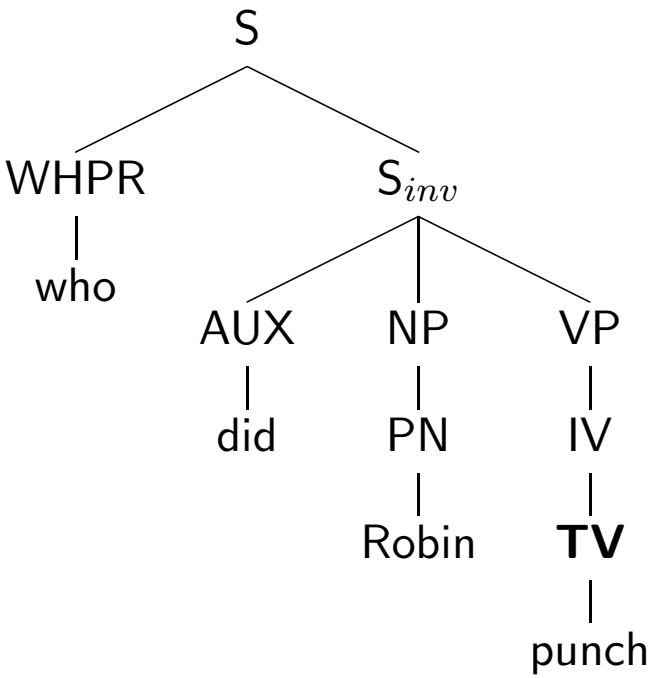
Questions

Other interrogatives involve long-distance dependencies

- We must record the presence of the missing phrase
- Revise the $\lambda-$term accordingly
- Link the variable of the missing phrase to the wh-word variable

(8) a. Who did Robin punch?

b. $?_x(person(x) \wedge punch(robin, x))$



Questions

Lexical rules + a third slot for wh-variables

(assuming the simplified $\lambda-$terms used in the Prolog implementation)

- All verbs are listed as having an empty wh-variable slot, e.g.
  (IV; $\lambda x.sneeze(x)$, [ ]) $\rightarrow$ sneezed
  (TV; $\lambda x.\lambda y.sneeze(x,y)$, [ ]) $\rightarrow$ punched

- One rule for complement interrogatives
  (IV; $\lambda x.\phi$, [y]) $\rightarrow$ (TV; $\lambda x.\lambda y.\phi$, [ ])

  Example:
  (TV ; $\lambda x.\lambda.y.punch(x,y)$, []) becomes (IV ; $\lambda x.punch(x,y)$, [y])

- One rule for subject interrogatives
  (TV; $\lambda y.\phi$, [x]) $\rightarrow$ (TV; $\lambda x.\lambda y.\phi$, [ ])

  Example:
  (TV; $\lambda x.\lambda y.punch(x,y)$, []) becomes (TV; $\lambda y.punch(x,y)$, [x])

Questions

Revised verbal rules

1. `rule(vp(X^K,[]),[tv(X^Y,[]),np(Y^K)])`
2. `rule(vp(X,WH),[iv(X,WH)])`
3. `rule(s(Y,WH),[np(X^Y),vp(X,WH)])`

New verbal rules

1. `rule(vp(K,[WH]),[tv(Y,[WH]),np(Y^K)])`
2. `rule(s(X,[WH]),[vp(X,[WH])])`

Wh-question rules:

1. `rule(Y,[whpr(X^Y),vp(X,[])])`
2. `rule(ynq(Y),[aux, np(X^Y),vp(X,[])])`
3. `rule(Z,[whpr((X^Y)^Z), inv_s(Y,[X])])`
4. `rule(inv_s(Y,[WH]),[aux, np(X^Y),vp(X,[WH])])`

Questions

Now our system can cope with questions:

```
?- parse([who,saw,tom]).
q(_G18,and(person(_G18),see(_G18,tom)))
true.

?- parse([who,did,tom,see]).
q(_G20,and(person(_G20),see(tom,_G20)))
true.

?- parse([who,tom,see]).
false.

?- parse([who,tom,saw]).
false.
```

Questions

PP Complements

(9) a. Who did John rely on?
$?_x(person(x), rely(john, x))$

b. What did John dispose of?
$?_x(thing(x), dispose(john, x))$

c. Who did John talk to?
$?_x(person(x), talk(john, x))$

d. What is John looking at?
$?_x(thing(x), look(john, x))$

Such case-marking prepositions add no semantics

(P; $\lambda P.P$, [ ]) → on | of | to | at | ...

(PP; $\lambda P.P(x)$, [x]) → on | of | to | at | ...

New rules (PV= verbs that require PP complements):

(PV; $\lambda x.\lambda y.rely(x, y)$, []) → rely

(VP; ..., WH) → (PV; ...) (PP; ... WH)

(semantics left out as an exercise)

Questions

The wh-store allows us to capture other patterns:

(10) a. Who did Robin [[punch _] and [kick _]]?

b. Who did Robin say [[John punched _] and [Sue kicked _]]?

c. What do most geneticists [[think about _] but [never consider the implications of _]]?

Generalization: coordination unifies wh-slots!

$(X\ ;\ ((\phi)(\psi))(\theta);\ WH) \rightarrow (X;\ \theta\ ;\ WH)$ (Coord; ϕ) $(X;\ \psi\ ;\ WH)$

Where Xs are variables over categories: NP, VP, S, PP, etc.

(Coord ; $\lambda P.\lambda Q.P \wedge Q) \rightarrow$ and | but
(Coord ; $\lambda P.\lambda Q.P \vee Q) \rightarrow$ or

Questions

Relative clauses also make use of the slot:

(11) a. The man [[who] [John punched _]] sued the restaurant.

b. A spy intercepted the book [[which] [the Canadian agent gave _]] the Mexican ambassador.

c. The informant knows [[who] [the email accused _]].

(12) a. The man [[who] [punched John]] sued the restaurant.

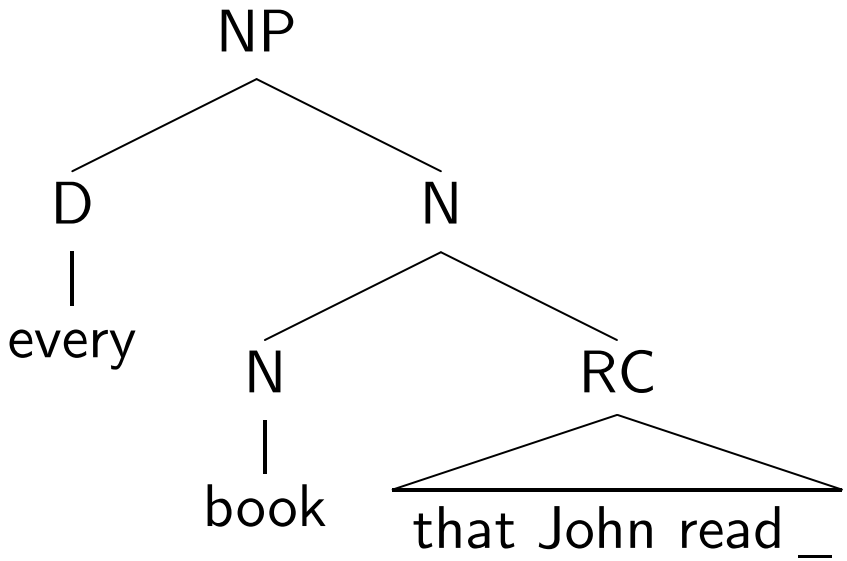
b. The informant knows [[who] [wrote the email to the ambassador]].

RCs syntax and semantics:

- (RC; ϕ, [x]) $\rightarrow$ REL (S; ϕ, [x])
- REL $\rightarrow$ that | what | who | which

Questions

Let us simplify things and assume that RCs combine with N:



Should yield:

$\lambda Q.\forall x((book(x) \wedge read(john, x)) \Rightarrow Q(x))$

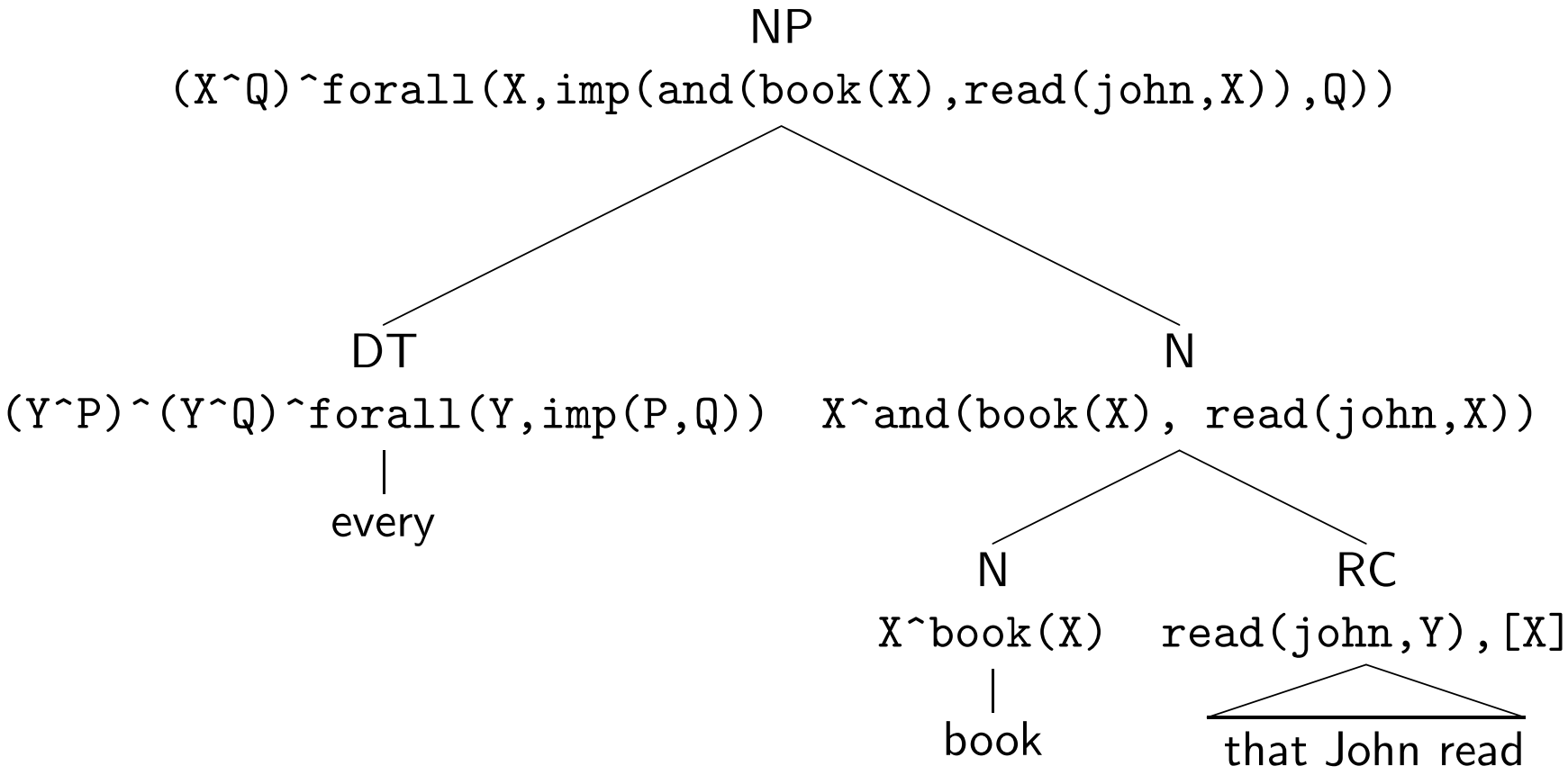
We can achieve this with the following rules:

1. `rule(n(X^and(Y,Z)),[n(X^Y),rc(X^Z,[])])`
2. `rule(n(X^and(Y,Z)),[n(X^Y),rc(Z,[X])])`

Questions

Example 1: ‘every book that John read’

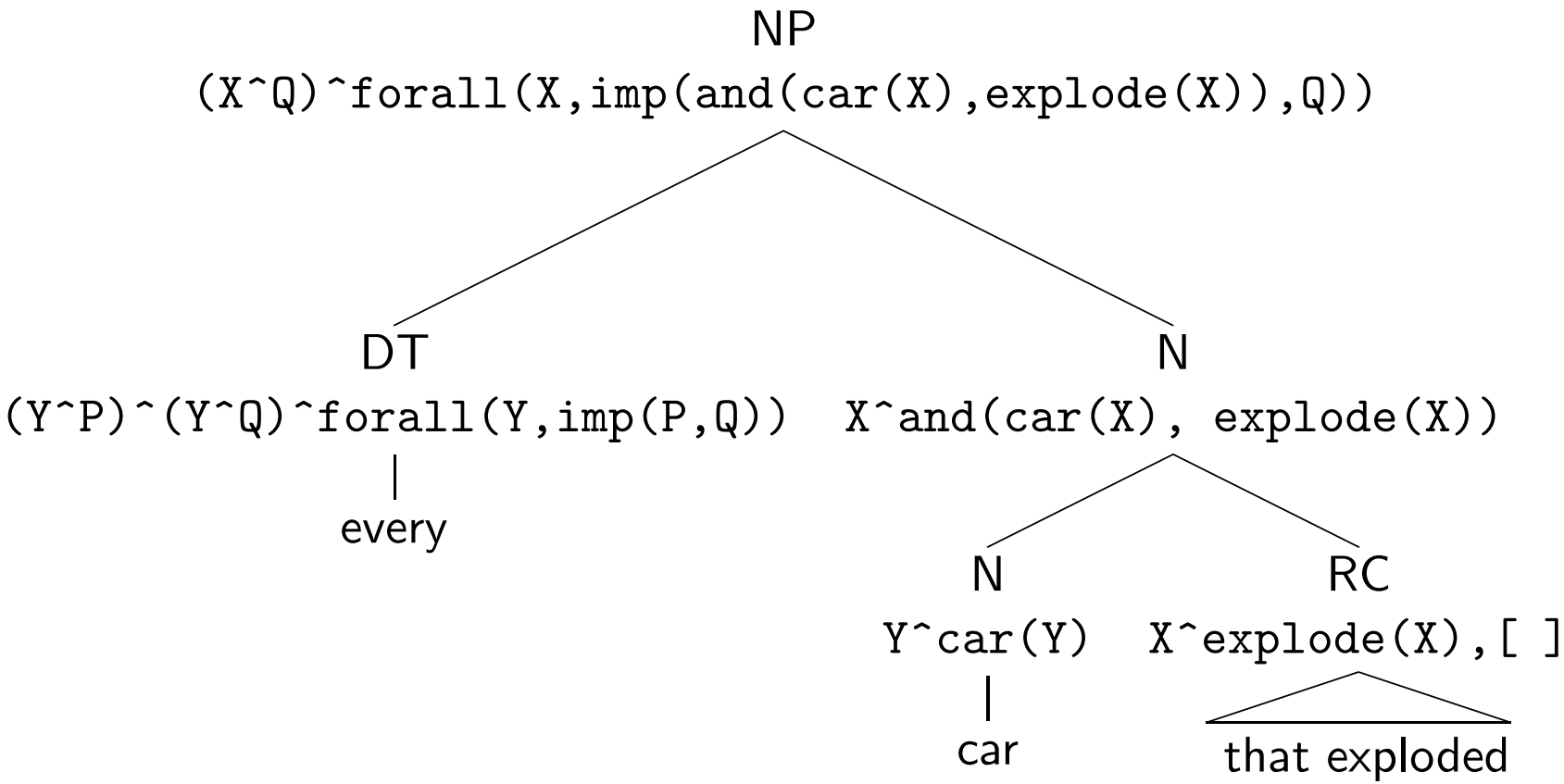
`rule(n(X^and(Y,Z)),[n(X^Y),rc(Z,[X])])`



Questions

Example 2: 'every car that exploded'

```
rule(n(X^and(Y,Z)),[n(X^Y),rc(X^Z,[])])
```



Questions

Deeper RC gap embeddings:

(13) a. Chelsea and David are two of the most amazing people that [we have ever had [the chance to work with _]].
(http://eventsbykristin.blogspot.com/2012_06_01_archive.html)

b. That is a good question [that I don't have [an answer for _]].
(http://www.recumbentriders.org/forums/showthread.php?t=4440)