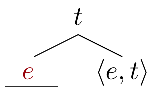
F13 Type composition exercise

Course in Semantics · Ling 531 / 731
McKenzie · University of Kansas

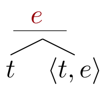
Key

Given the structure, fill in the missing type. Note that this is an abstract exercise. The structures might not correspond to anything we see. The point is to work with types, whatever they happen to be.

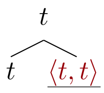
Ex:



Ex:



Ex:

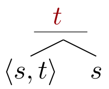


1 Figuring out the output type

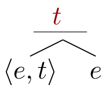
In each instance, figure out the type of the result of combining the two listed nodes by functional application.

1.1 simple type argument

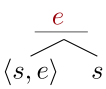
1.



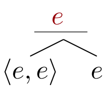
2.



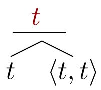
3.



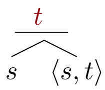
4.



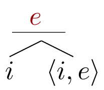
5.



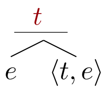
6.



7.

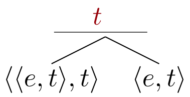


8.

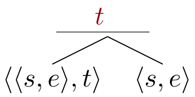


1.2 function arguments (i.e. functions as arguments)

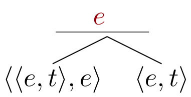
1.



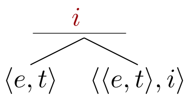
2.



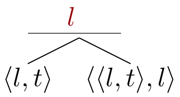
3.



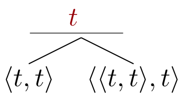
4.



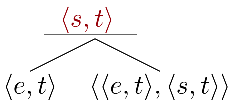
5.



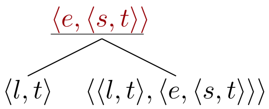
6.



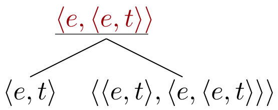
7.



8.


$\langle e,\langle s,t\rangle\rangle$
$\langle l,t\rangle$
$\langle\langle l,t\rangle,\langle e,\langle s,t\rangle\rangle\rangle$

9.

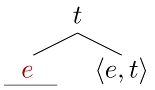

$\langle e,\langle e,t\rangle\rangle$
$\langle e,t\rangle$
$\langle\langle e,t\rangle,\langle e,\langle e,t\rangle\rangle\rangle$

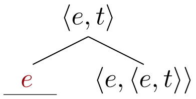
2 Figuring out input types

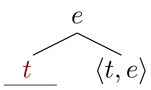
In each instance, figure out the type of one of the inputs by deducing from the listed nodes.

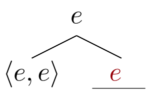
2.1 Simple-type argument

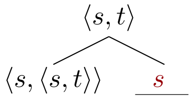
All the answers here are of simple type, even if complex types are technically possible.

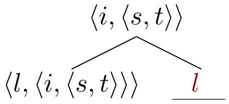
1.


2.


3.


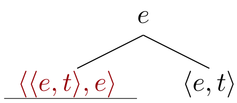
4.


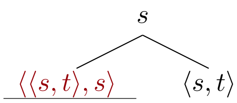
5.


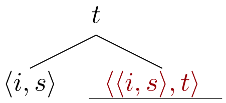
6.

$\langle i,\langle s,t\rangle\rangle$
$\langle l,\langle i,\langle s,t\rangle\rangle\rangle$
l

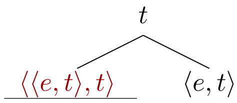
2.2 Complex-type argument

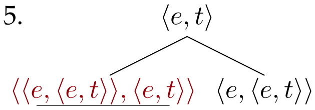
All of the answers here are of complex type $(\langle\sigma,\tau\rangle)$, even if simple types are technically possible.

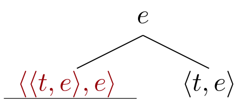
1.


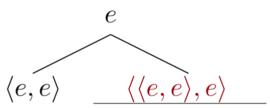
2.


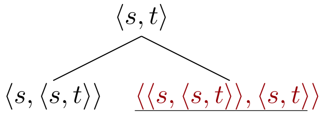
3.


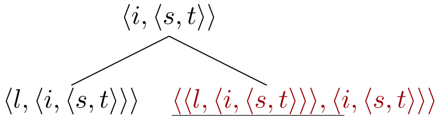
4.


5.

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

6.


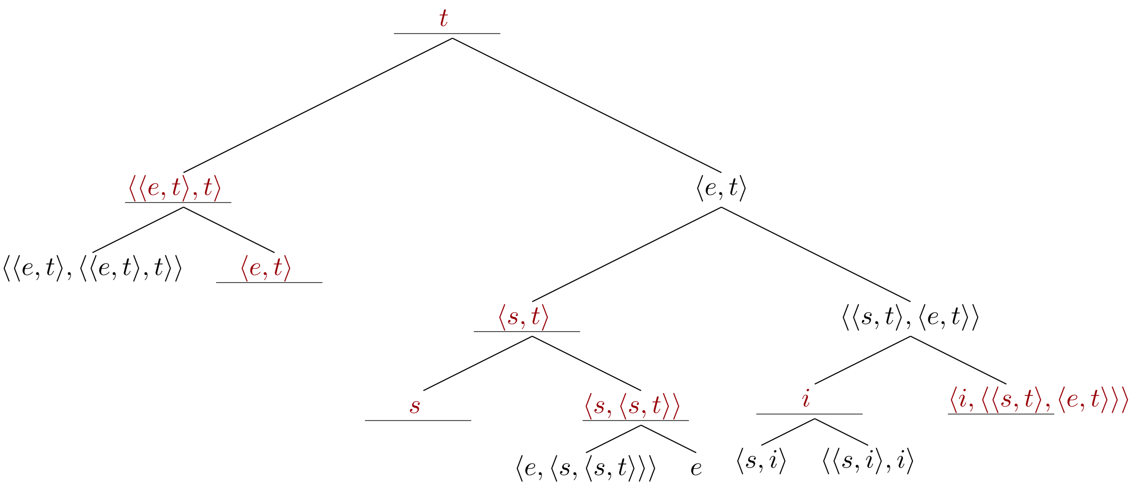
7.


8.

$\langle s,t\rangle$
$\langle s,\langle s,t\rangle\rangle$
$\langle\langle s,\langle s,t\rangle\rangle,\langle s,t\rangle\rangle$

9.

$\langle i,\langle s,t\rangle\rangle$
$\langle l,\langle i,\langle s,t\rangle\rangle\rangle$
$\langle\langle l,\langle i,\langle s,t\rangle\rangle\rangle,\langle i,\langle s,t\rangle\rangle\rangle$

3 Larger structure

Give this one a go.


t
$\langle\langle e,t\rangle,t\rangle$
$\langle e,t\rangle$
$\langle\langle e,t\rangle,\langle\langle e,t\rangle,t\rangle\rangle$
$\langle e,t\rangle$
$\langle s,t\rangle$
$\langle\langle s,t\rangle,\langle e,t\rangle\rangle$
s
$\langle s,\langle s,t\rangle\rangle$
i
$\langle i,\langle\langle s,t\rangle,\langle e,t\rangle\rangle\rangle$
$\langle e,\langle s,\langle s,t\rangle\rangle\rangle$
e
$\langle s,i\rangle$
$\langle\langle s,i\rangle,i\rangle$