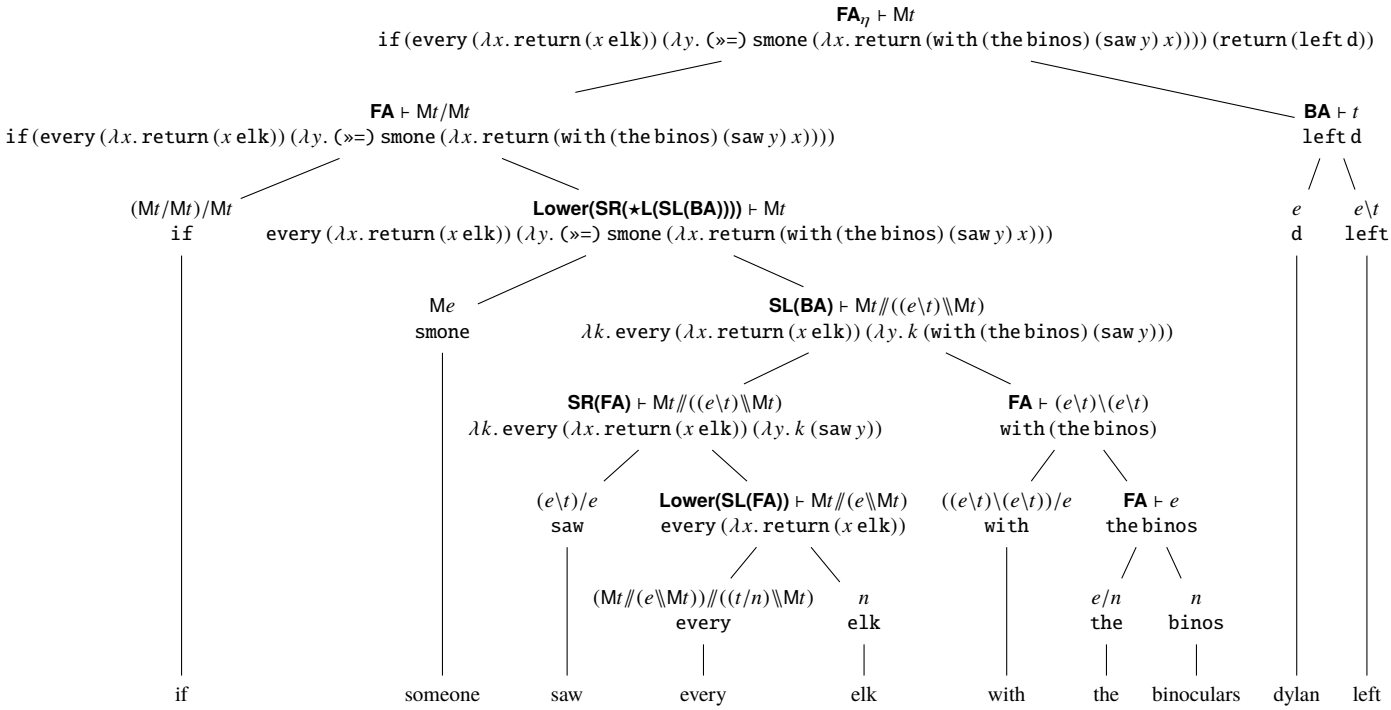
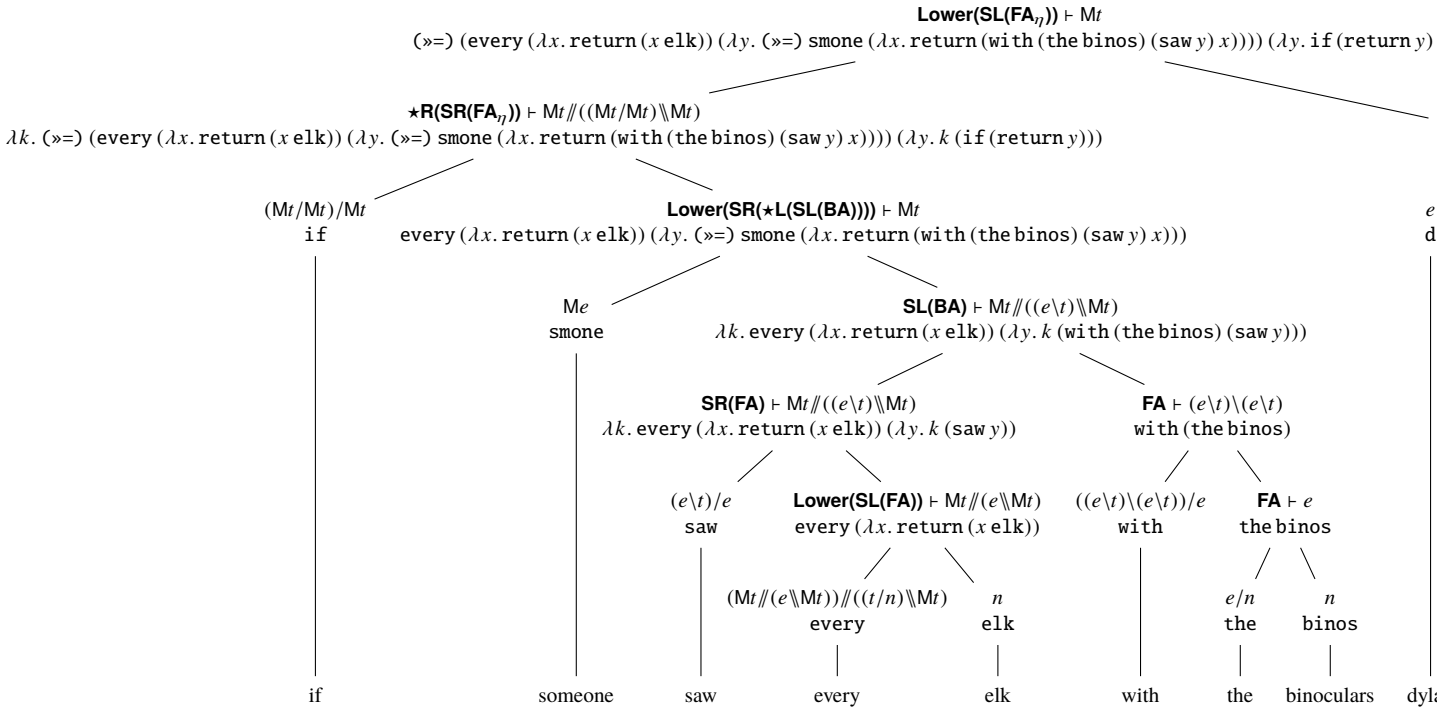
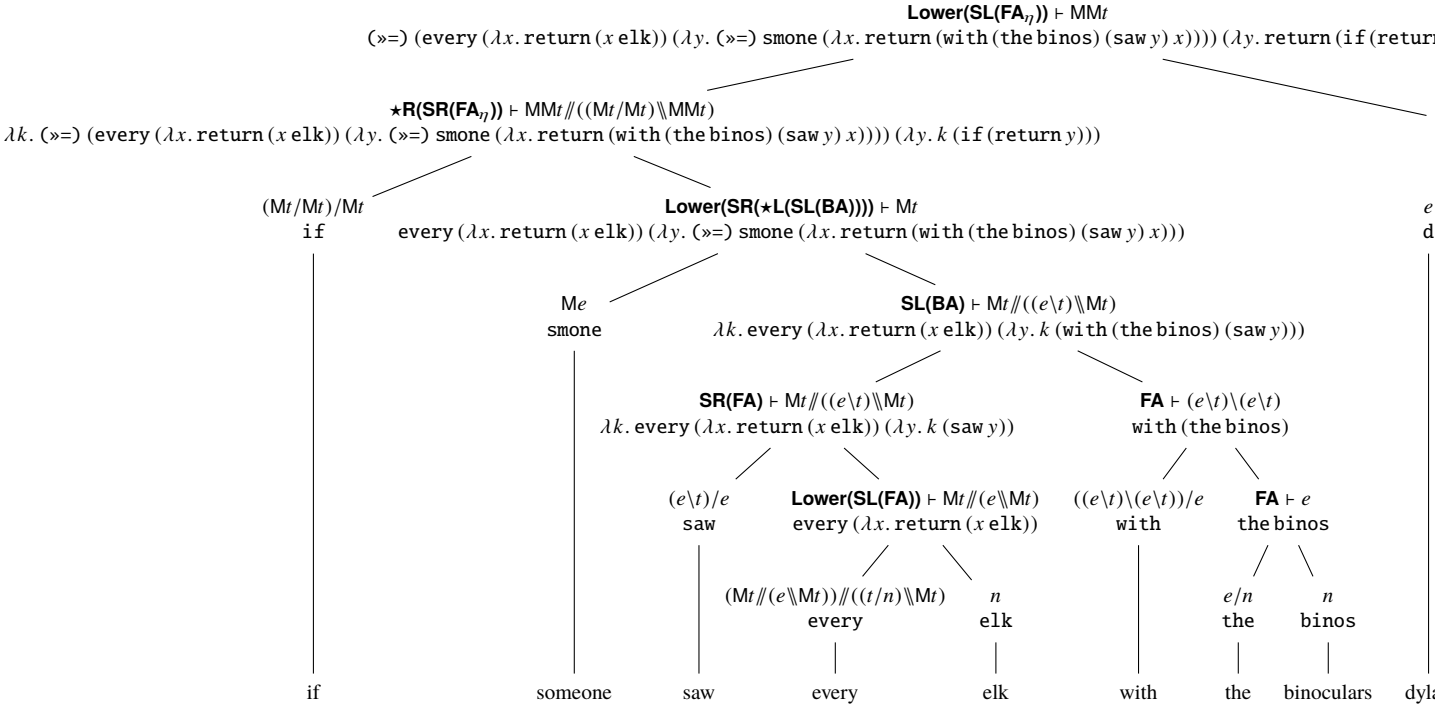
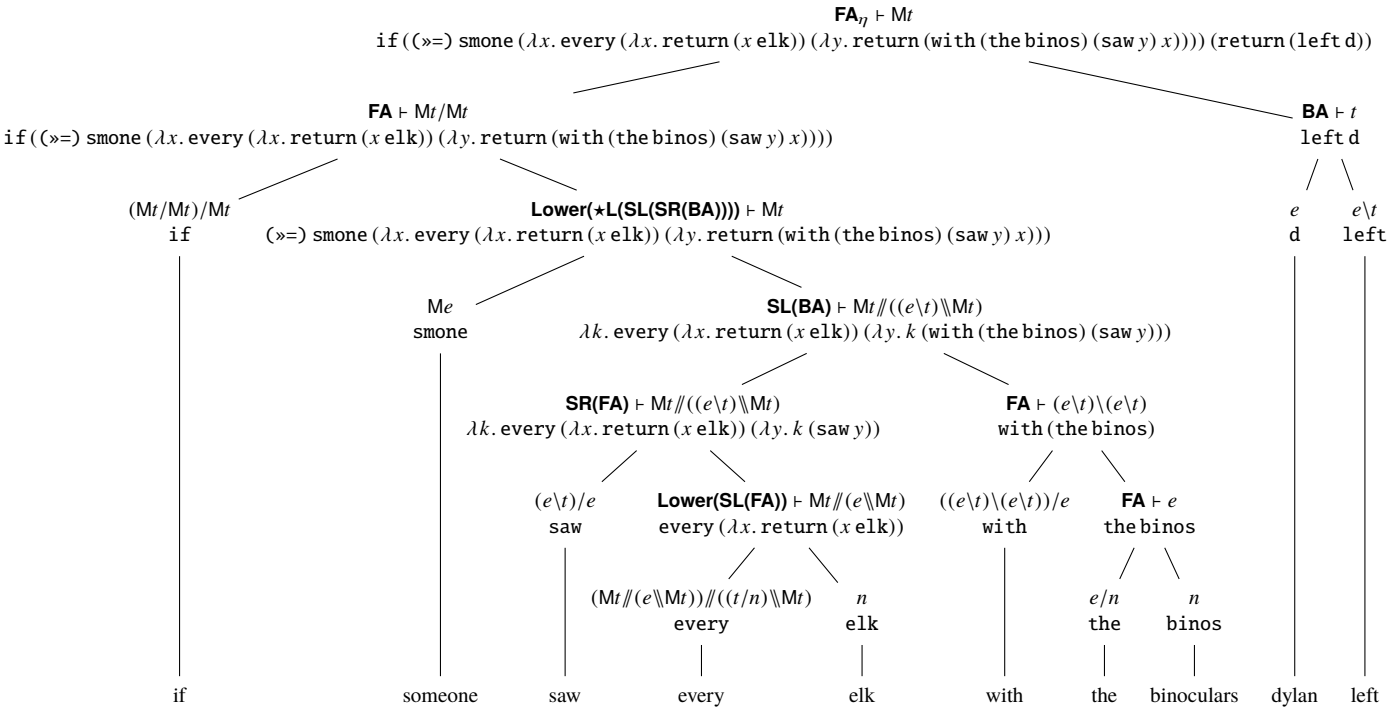
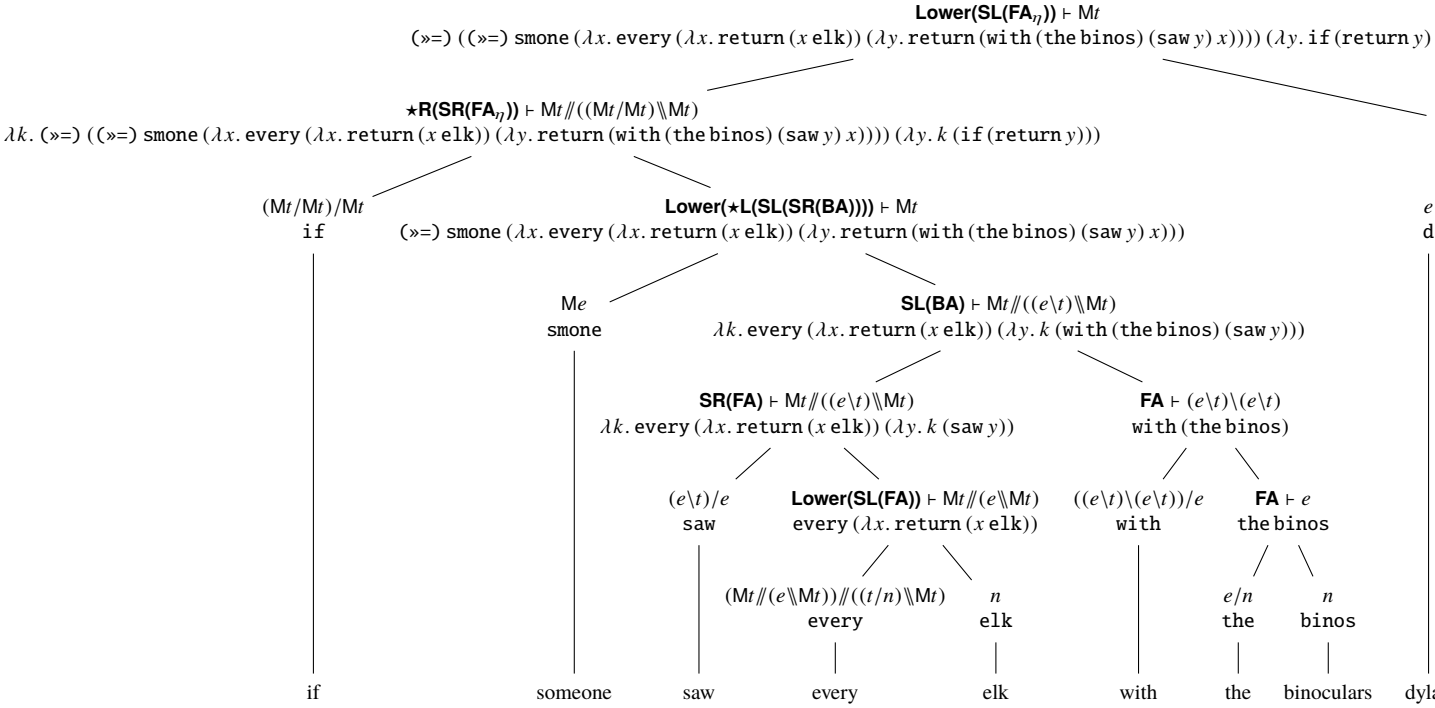
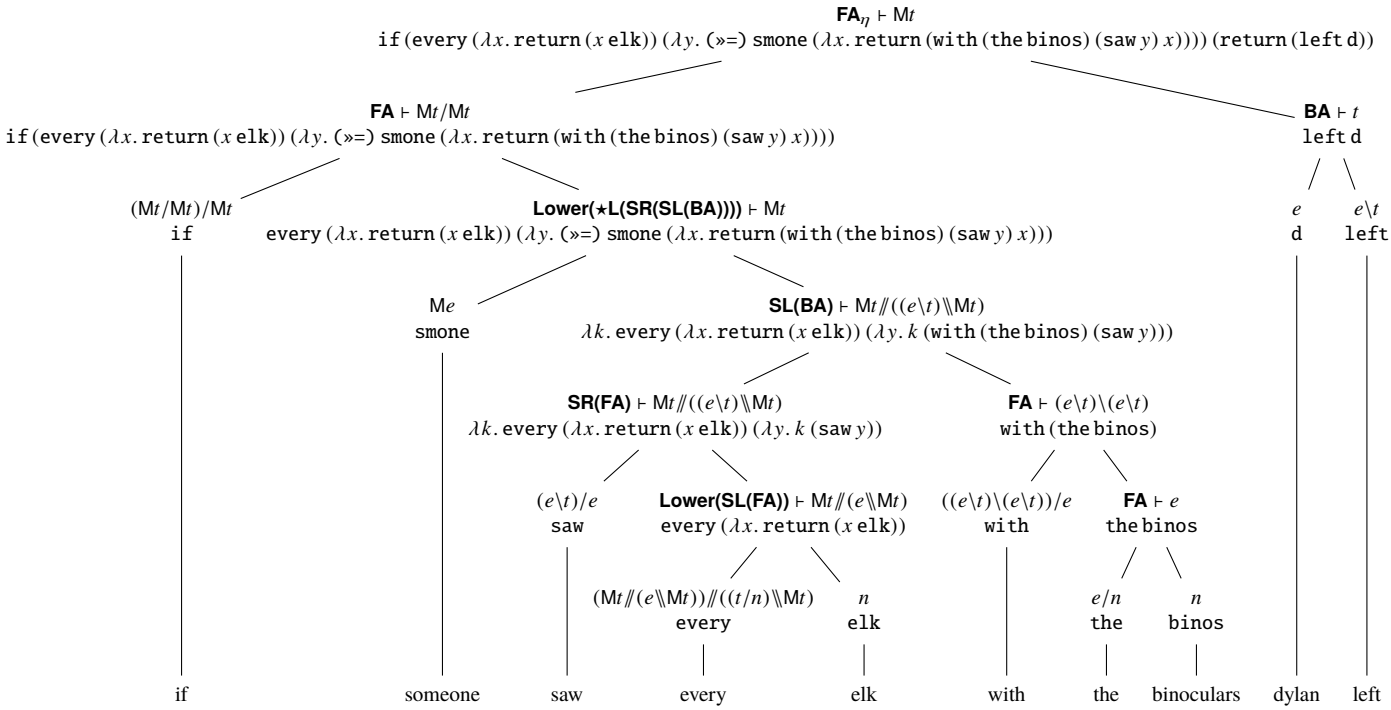

FA_η ⊢ Mt
if (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (return (left d))
FA ⊢ Mt/Mt
if (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x))))
BA ⊢ t
left d
(Mt/Mt)/Mt
if
Lower(SR(⋆L(SL(BA)))) ⊢ Mt
every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))
e
d
e\t
left
Me
smone
SL(BA) ⊢ Mt⫽((e\t)\\Mt)
λk. every (λx. return (x elk)) (λy. k (with (the binos) (saw y)))
SR(FA) ⊢ Mt⫽((e\t)\\Mt)
λk. every (λx. return (x elk)) (λy. k (saw y))
FA ⊢ (e\t)\(e\t)
with (the binos)
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt⫽(e\\Mt)
every (λx. return (x elk))
((e\t)\(e\t))/e
with
FA ⊢ e
the binos
(Mt⫽(e\\Mt))⫽((t/n)\\Mt)
every
n
elk
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
dylan
left

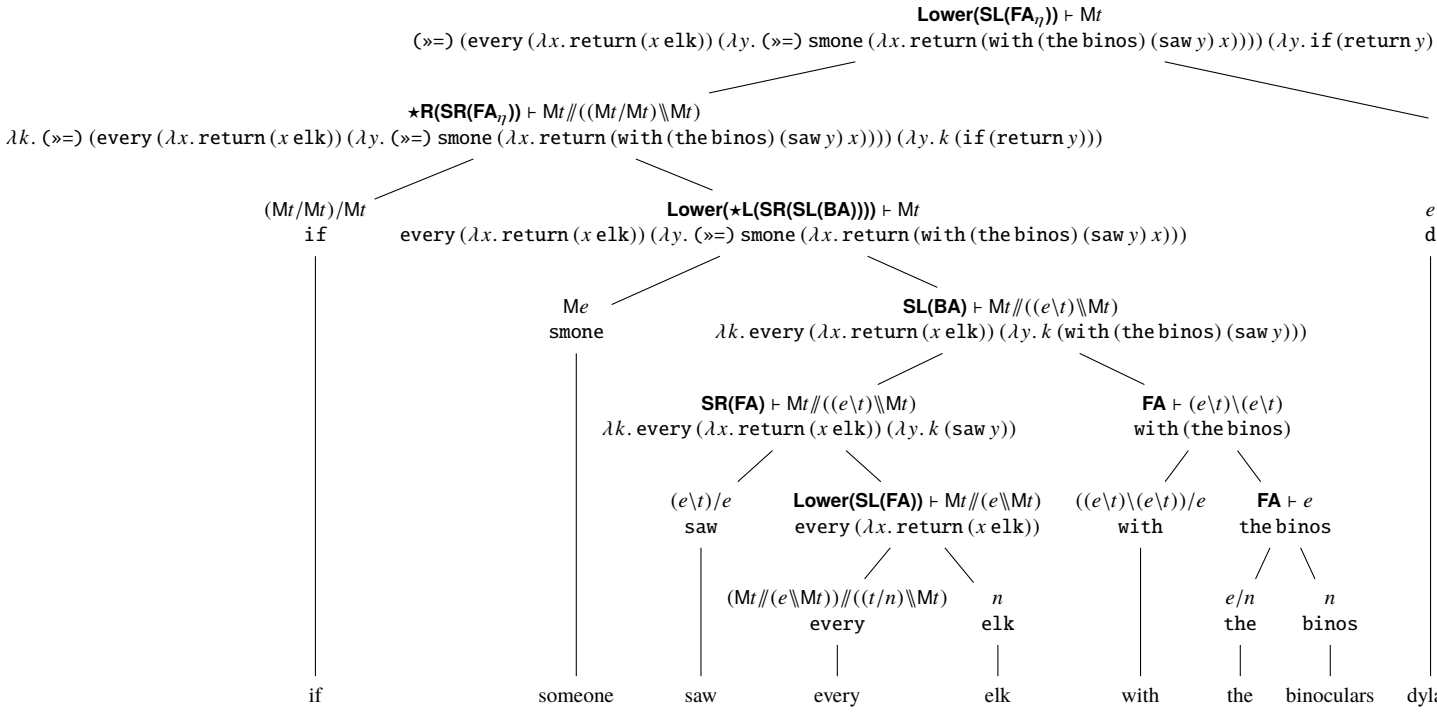

Lower(SL(FA_η)) ⊢ Mt
(»=) (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (λy. if (return y)
⋆R(SR(FA_η)) ⊢ Mt⫽((Mt/Mt)\\Mt)
λk. (»=) (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(SR(⋆L(SL(BA)))) ⊢ Mt
every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))
Me
smone
SL(BA) ⊢ Mt⫽((e\t)\\Mt)
λk. every (λx. return (x elk)) (λy. k (with (the binos) (saw y)))
SR(FA) ⊢ Mt⫽((e\t)\\Mt)
λk. every (λx. return (x elk)) (λy. k (saw y))
FA ⊢ (e\t)\(e\t)
with (the binos)
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt⫽(e\\Mt)
every (λx. return (x elk))
((e\t)\(e\t))/e
with
FA ⊢ e
the binos
(Mt⫽(e\\Mt))⫽((t/n)\\Mt)
every
n
elk
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
dyla

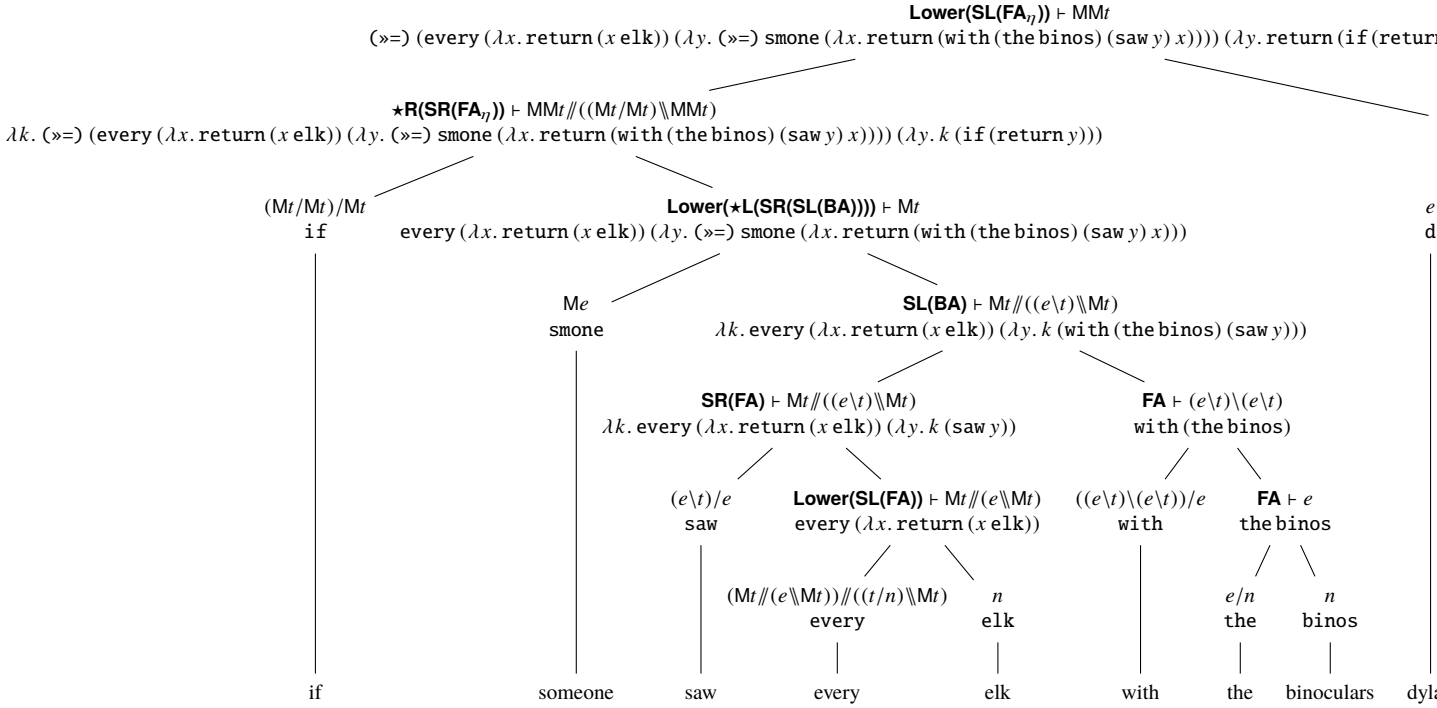

Lower(SL(FA$_\eta$)) ⊢ MMt
(»=) (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (λy. return (if (retur
★R(SR(FA$_\eta$)) ⊢ MMt // ((Mt/Mt)\\MM$t$)
λk. (»=) (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(SR(★L(SL(BA)))) ⊢ Mt
every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))
Me
smone
SL(BA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (with (the binos) (saw y)))
SR(FA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (saw y))
FA ⊢ (e\t)\(e\t)
with (the binos)
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt // (e\\M$t$)
every (λx. return (x elk))
((e\t)\(e\t))/e
with
FA ⊢ e
the binos
(Mt // (e\\M$t$)) // ((t/n)\\M$t$)
every
n
elk
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
dyla

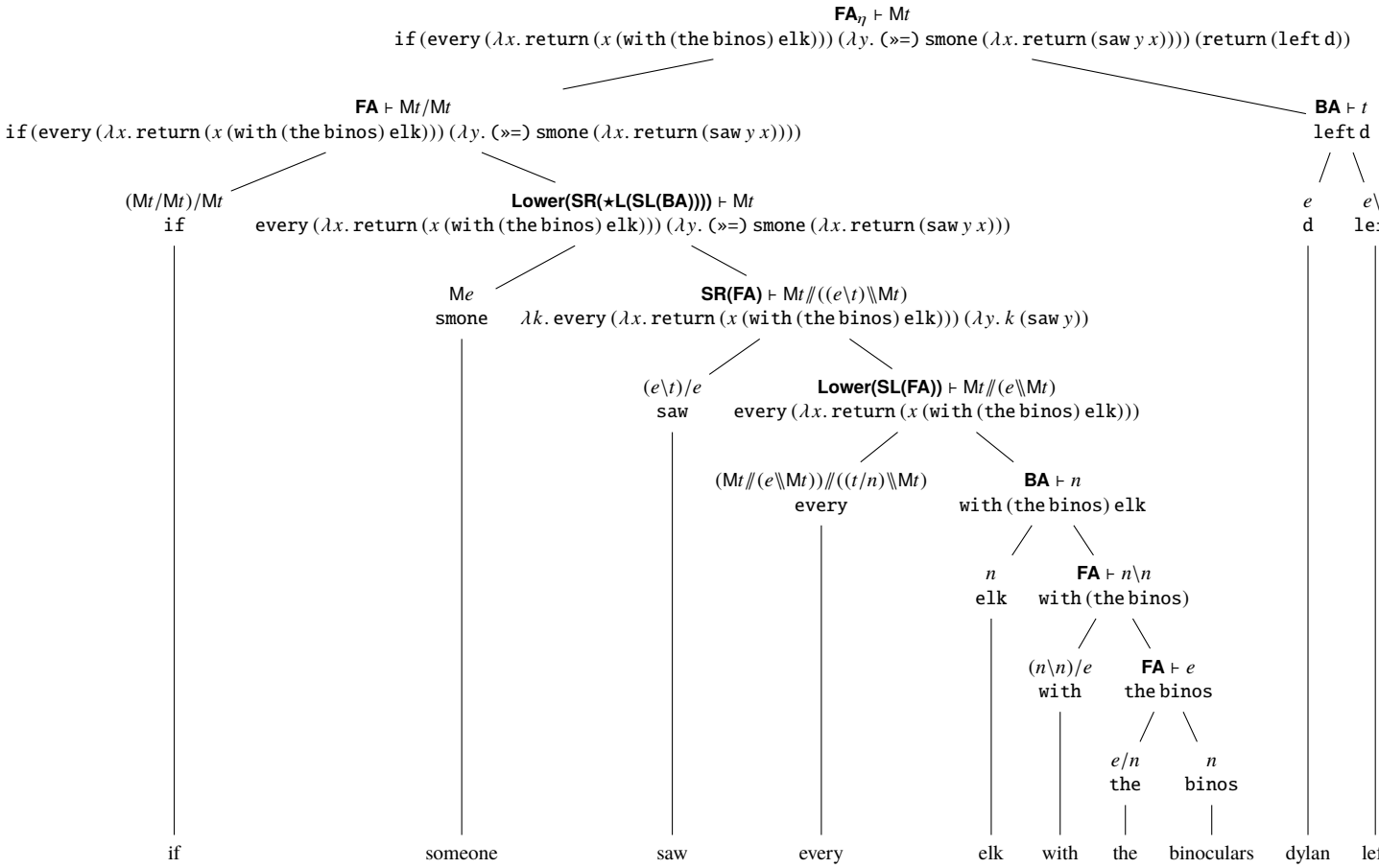

FA$_\eta$ ⊢ Mt
if ((»=) smone (λx. every (λx. return (x elk)) (λy. return (with (the binos) (saw y) x)))) (return (left d))
FA ⊢ Mt/Mt
if ((»=) smone (λx. every (λx. return (x elk)) (λy. return (with (the binos) (saw y) x))))
BA ⊢ t
left d
(Mt/Mt)/Mt
if
Lower(★L(SL(SR(BA)))) ⊢ Mt
(»=) smone (λx. every (λx. return (x elk)) (λy. return (with (the binos) (saw y) x)))
e
d
e\t
left
Me
smone
SL(BA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (with (the binos) (saw y)))
SR(FA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (saw y))
FA ⊢ (e\t)\(e\t)
with (the binos)
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt // (e\\M$t$)
every (λx. return (x elk))
((e\t)\(e\t))/e
with
FA ⊢ e
the binos
(Mt // (e\\M$t$)) // ((t/n)\\M$t$)
every
n
elk
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
dylan
left

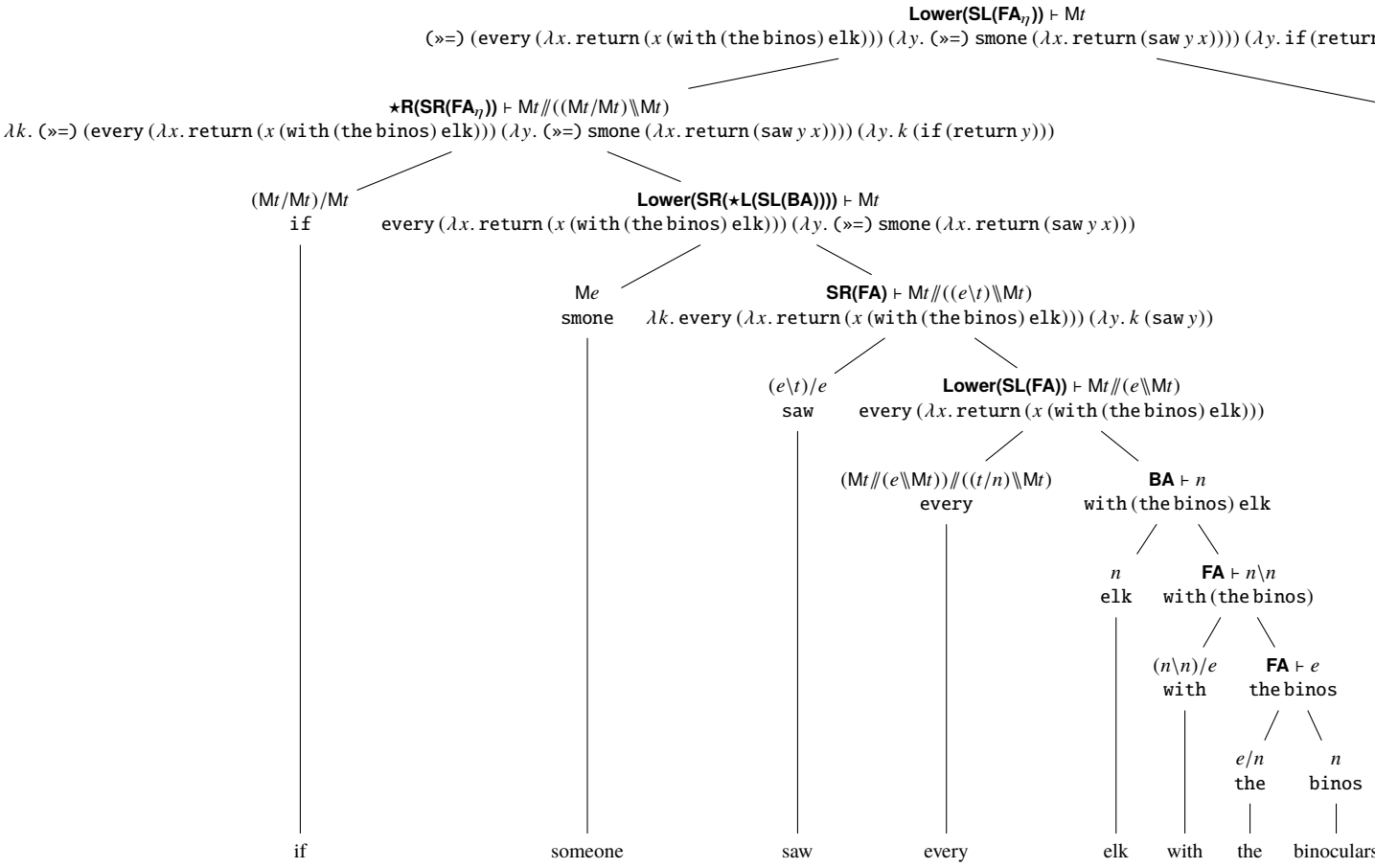

Lower(SL(FA$_\eta$)) ⊢ Mt
(»=) ((»=) smone (λx. every (λx. return (x elk)) (λy. return (with (the binos) (saw y) x)))) (λy. if (return y)
★R(SR(FA$_\eta$)) ⊢ Mt⫽((Mt/Mt)\\M$t$)
λk. (»=) ((»=) smone (λx. every (λx. return (x elk)) (λy. return (with (the binos) (saw y) x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(★L(SL(SR(BA)))) ⊢ Mt
(»=) smone (λx. every (λx. return (x elk)) (λy. return (with (the binos) (saw y) x)))
Me
smone
SL(BA) ⊢ Mt⫽((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (with (the binos) (saw y)))
SR(FA) ⊢ Mt⫽((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (saw y))
FA ⊢ (e\t)\(e\t)
with (the binos)
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt⫽(e\\M$t$)
every (λx. return (x elk))
((e\t)\(e\t))/e
with
FA ⊢ e
the binos
(Mt⫽(e\\M$t$))⫽((t/n)\\M$t$)
every
n
elk
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
Lower(SL(FA$_\eta$)) ⊢ MMt
(»=) ((»=) smone (λx. every (λx. return (x elk)) (λy. return (with (the binos) (saw y) x)))) (λy. return (if (retur
★R(SR(FA$_\eta$)) ⊢ MMt⫽((Mt/Mt)\\MM$t$)
λk. (»=) ((»=) smone (λx. every (λx. return (x elk)) (λy. return (with (the binos) (saw y) x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(★L(SL(SR(BA)))) ⊢ Mt
(»=) smone (λx. every (λx. return (x elk)) (λy. return (with (the binos) (saw y) x)))
Me
smone
SL(BA) ⊢ Mt⫽((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (with (the binos) (saw y)))
SR(FA) ⊢ Mt⫽((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (saw y))
FA ⊢ (e\t)\(e\t)
with (the binos)
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt⫽(e\\M$t$)
every (λx. return (x elk))
((e\t)\(e\t))/e
with
FA ⊢ e
the binos
(Mt⫽(e\\M$t$))⫽((t/n)\\M$t$)
every
n
elk
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars

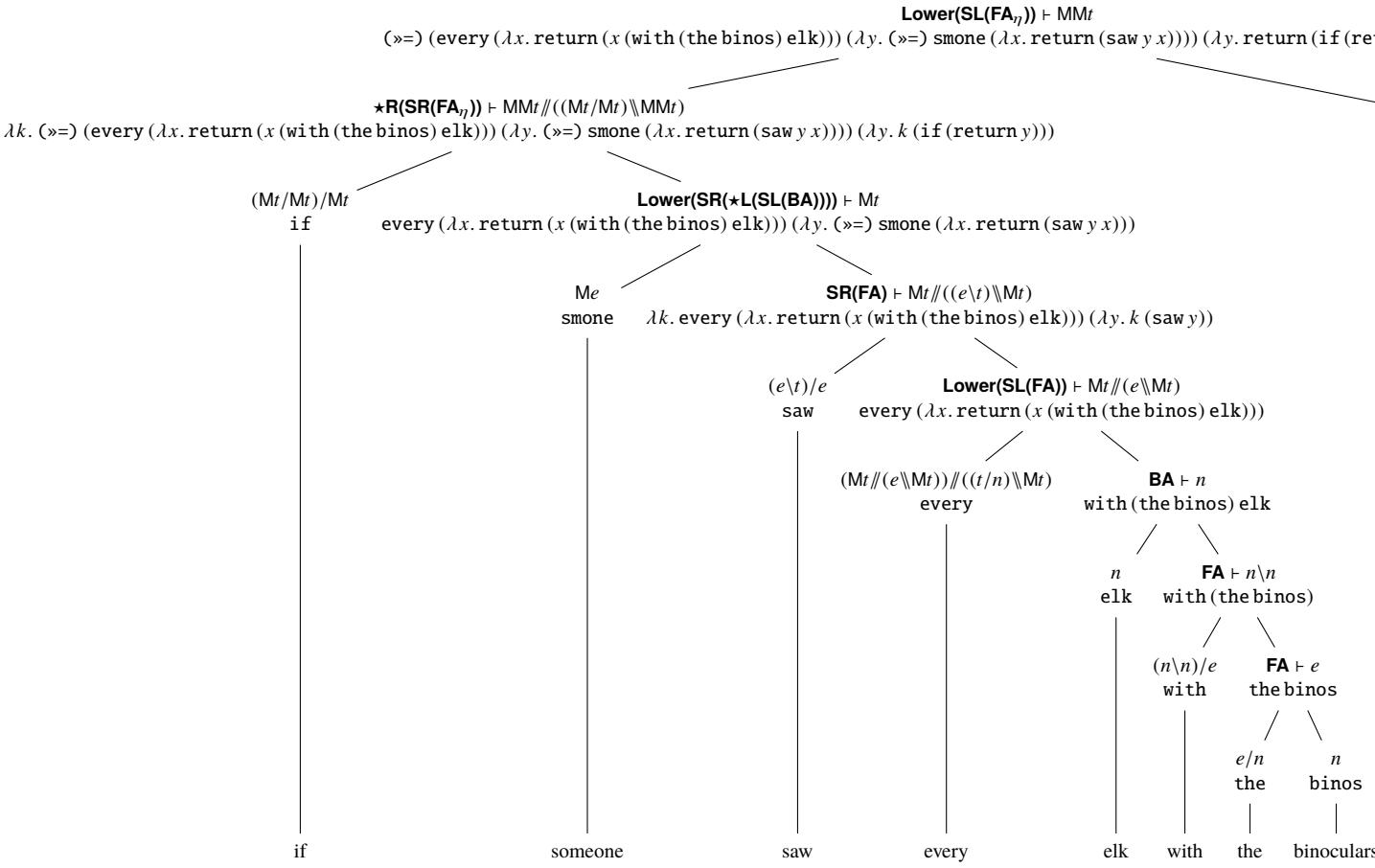

FA_η ⊢ Mt
if (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (return (left d))
FA ⊢ Mt/Mt
if (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x))))
BA ⊢ t
left d
(Mt/Mt)/Mt
if
Lower(⋆L(SR(SL(BA)))) ⊢ Mt
every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))
e
d
e\t
left
Me
smone
SL(BA) ⊢ Mt⫽((e\t)\\Mt)
λk. every (λx. return (x elk)) (λy. k (with (the binos) (saw y)))
SR(FA) ⊢ Mt⫽((e\t)\\Mt)
λk. every (λx. return (x elk)) (λy. k (saw y))
FA ⊢ (e\t)\(e\t)
with (the binos)
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt⫽(e\\Mt)
every (λx. return (x elk))
((e\t)\(e\t))/e
with
FA ⊢ e
the binos
(Mt⫽(e\\Mt))⫽((t/n)\\Mt)
every
n
elk
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
dylan
left

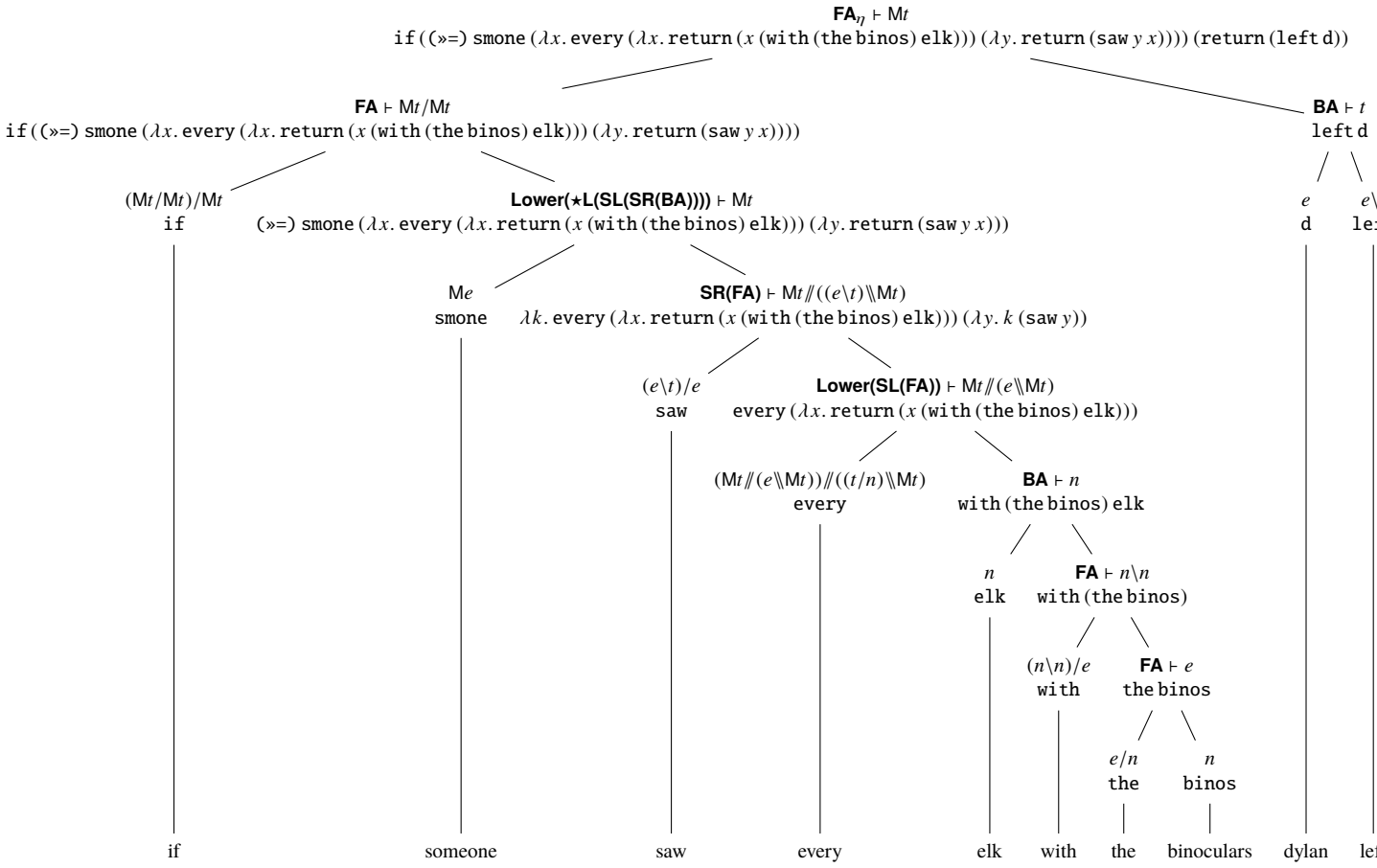

Lower(SL(FA_η)) ⊢ Mt
(»=) (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (λy. if (return y)
⋆R(SR(FA_η)) ⊢ Mt⫽((Mt/Mt)\\Mt)
λk. (»=) (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(⋆L(SR(SL(BA)))) ⊢ Mt
every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))
Me
smone
SL(BA) ⊢ Mt⫽((e\t)\\Mt)
λk. every (λx. return (x elk)) (λy. k (with (the binos) (saw y)))
SR(FA) ⊢ Mt⫽((e\t)\\Mt)
λk. every (λx. return (x elk)) (λy. k (saw y))
FA ⊢ (e\t)\(e\t)
with (the binos)
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt⫽(e\\Mt)
every (λx. return (x elk))
((e\t)\(e\t))/e
with
FA ⊢ e
the binos
(Mt⫽(e\\Mt))⫽((t/n)\\Mt)
every
n
elk
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars

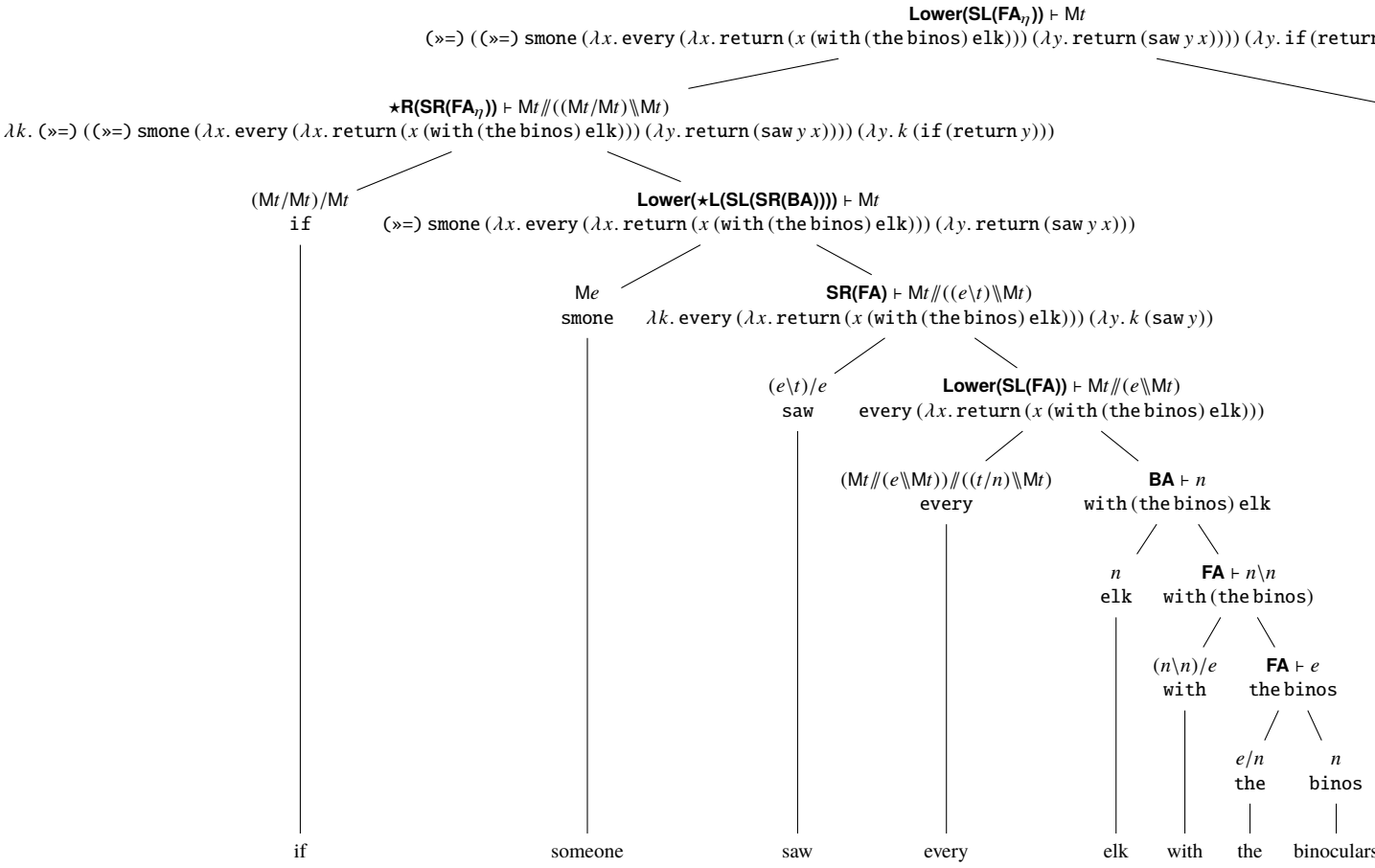

Lower(SL(FA$_\eta$)) ⊢ MMt
(»=) (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (λy. return (if (retur
★R(SR(FA$_\eta$)) ⊢ MMt // ((Mt/Mt)\\MM$t$)
λk. (»=) (every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(★L(SR(SL(BA)))) ⊢ Mt
every (λx. return (x elk)) (λy. (»=) smone (λx. return (with (the binos) (saw y) x)))
Me
smone
SL(BA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (with (the binos) (saw y)))
SR(FA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x elk)) (λy. k (saw y))
FA ⊢ (e\$t$)\($e$\$t$)
with (the binos)
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt // (e\\M$t$)
every (λx. return (x elk))
(((e\$t$)\($e$\$t$))/$e$
with
FA ⊢ e
the binos
(Mt // (e\\M$t$)) // ((t/n)\\M$t$)
every
n
elk
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
dyla

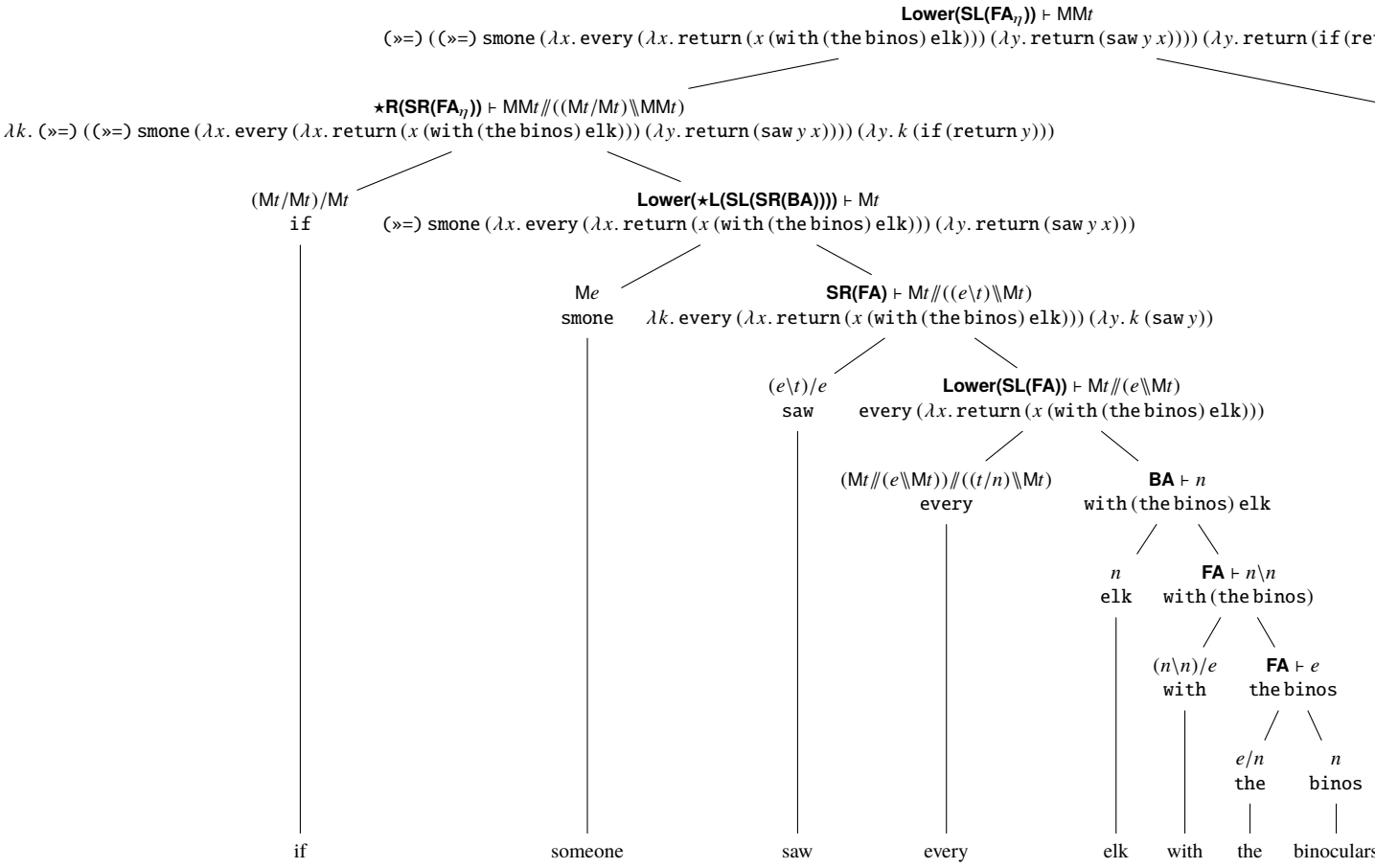

FA$_\eta$ ⊢ Mt
if (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (return (left d))
FA ⊢ Mt/Mt
if (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x))))
BA ⊢ t
left d
(Mt/Mt)/Mt
if
Lower(SR(★L(SL(BA)))) ⊢ Mt
every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))
Me
smone
SR(FA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x (with (the binos) elk))) (λy. k (saw y))
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt // (e\\M$t$)
every (λx. return (x (with (the binos) elk)))
(Mt // (e\\M$t$)) // ((t/n)\\M$t$)
every
BA ⊢ n
with (the binos) elk
n
elk
FA ⊢ n\n
with (the binos)
(n\n)/e
with
FA ⊢ e
the binos
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
dylan

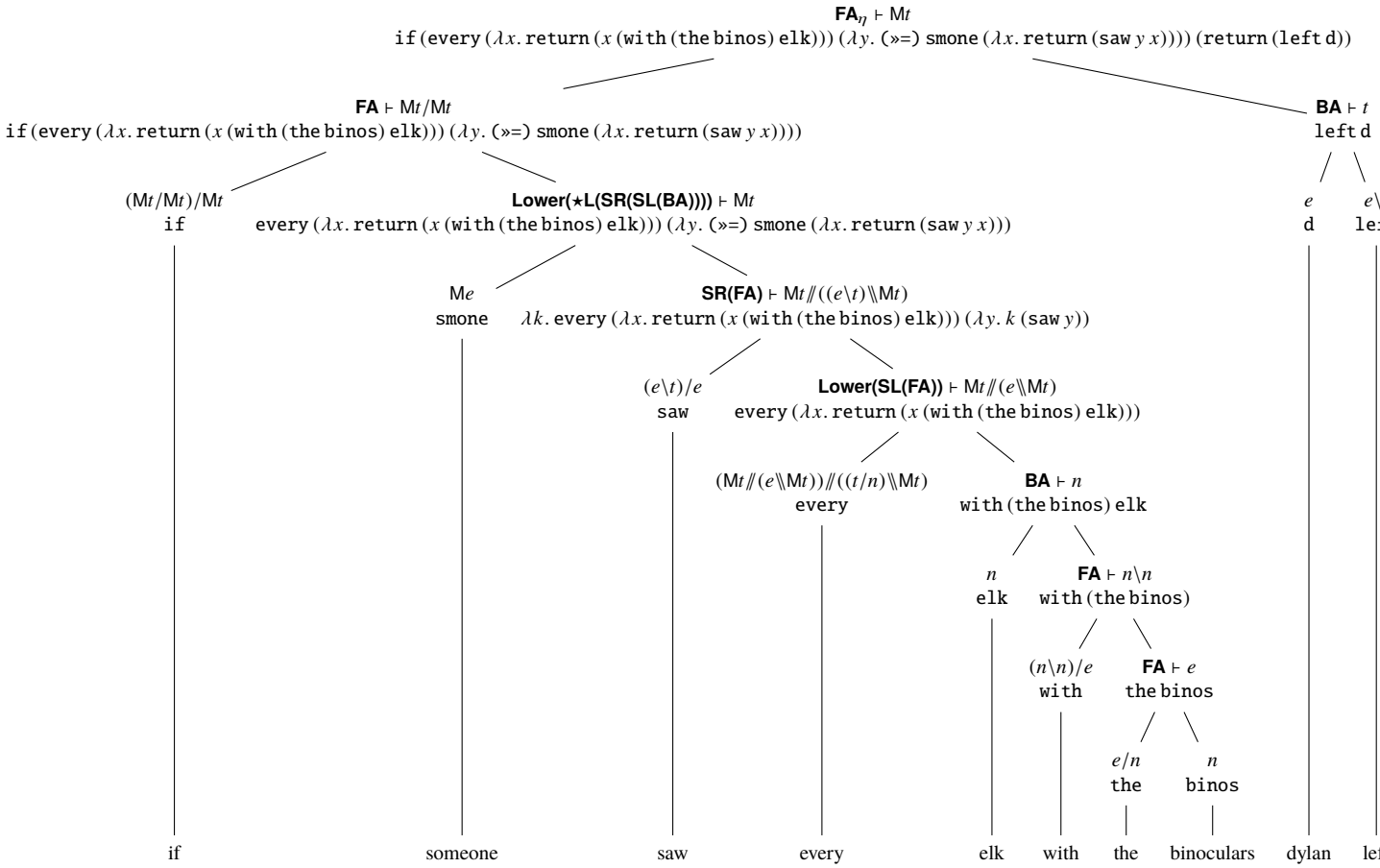

Lower(SL(FA$_\eta$)) ⊢ Mt
(»=) (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (λy. if (return
★R(SR(FA$_\eta$)) ⊢ Mt // ((Mt/Mt)\\M$t$)
λk. (»=) (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(SR(★L(SL(BA)))) ⊢ Mt
every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))
Me
smone
SR(FA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x (with (the binos) elk))) (λy. k (saw y))
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt // (e\\M$t$)
every (λx. return (x (with (the binos) elk)))
(Mt // (e\\M$t$)) // ((t/n)\\M$t$)
every
BA ⊢ n
with (the binos) elk
n
elk
FA ⊢ n\n
with (the binos)
(n\n)/e
with
FA ⊢ e
the binos
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars

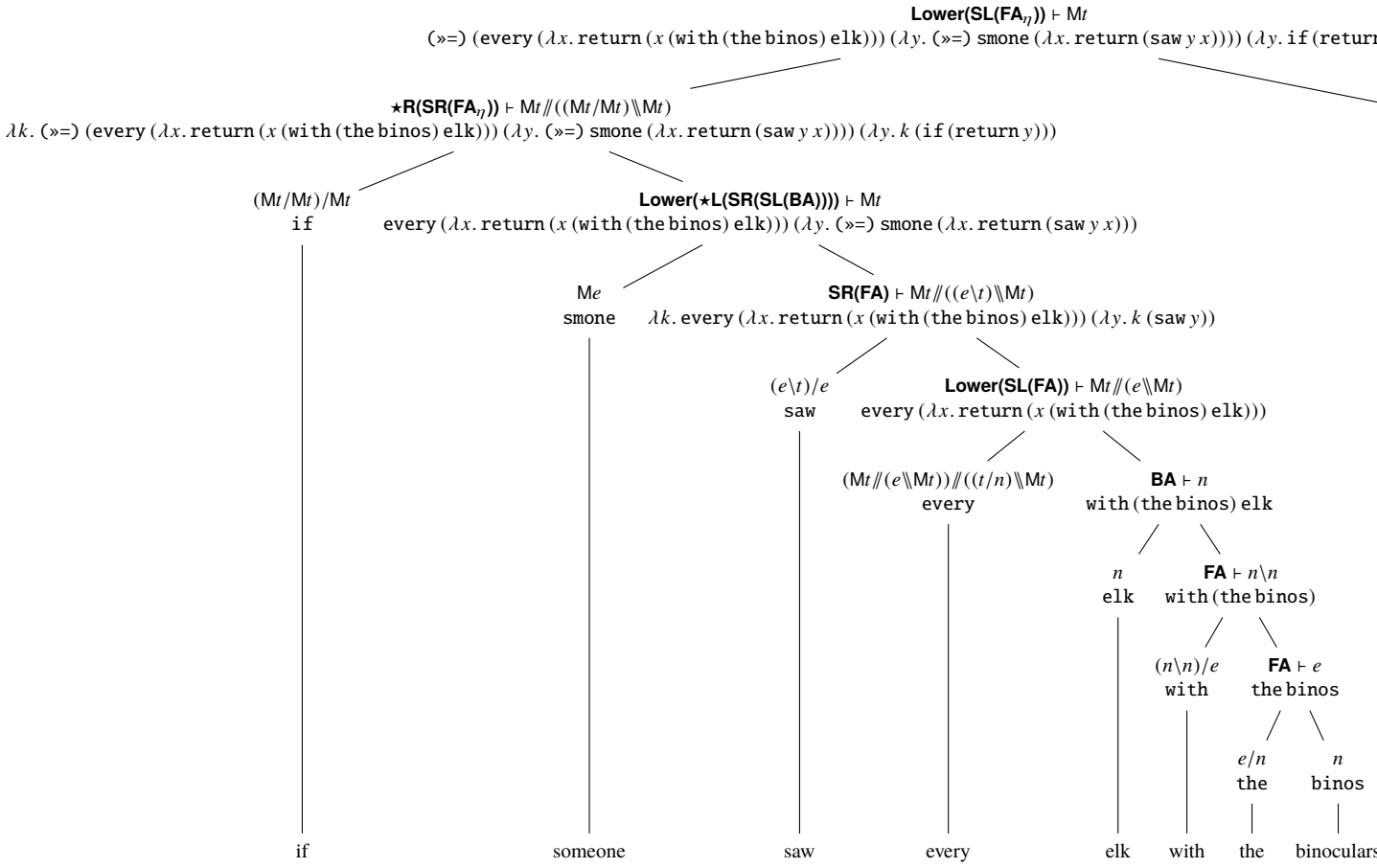

Lower(SL(FA$_\eta$)) ⊢ MMt
(»=) (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (λy. return (if (re
★R(SR(FA$_\eta$)) ⊢ MMt // ((Mt/Mt)\\MM$t$)
λk. (»=) (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(SR(★L(SL(BA)))) ⊢ Mt
every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))
Me
smone
SR(FA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x (with (the binos) elk))) (λy. k (saw y))
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt // (e\\M$t$)
every (λx. return (x (with (the binos) elk)))
(Mt // (e\\M$t$)) // ((t/n)\\M$t$)
every
BA ⊢ n
with (the binos) elk
n
elk
FA ⊢ n\n
with (the binos)
(n\n)/e
with
FA ⊢ e
the binos
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars

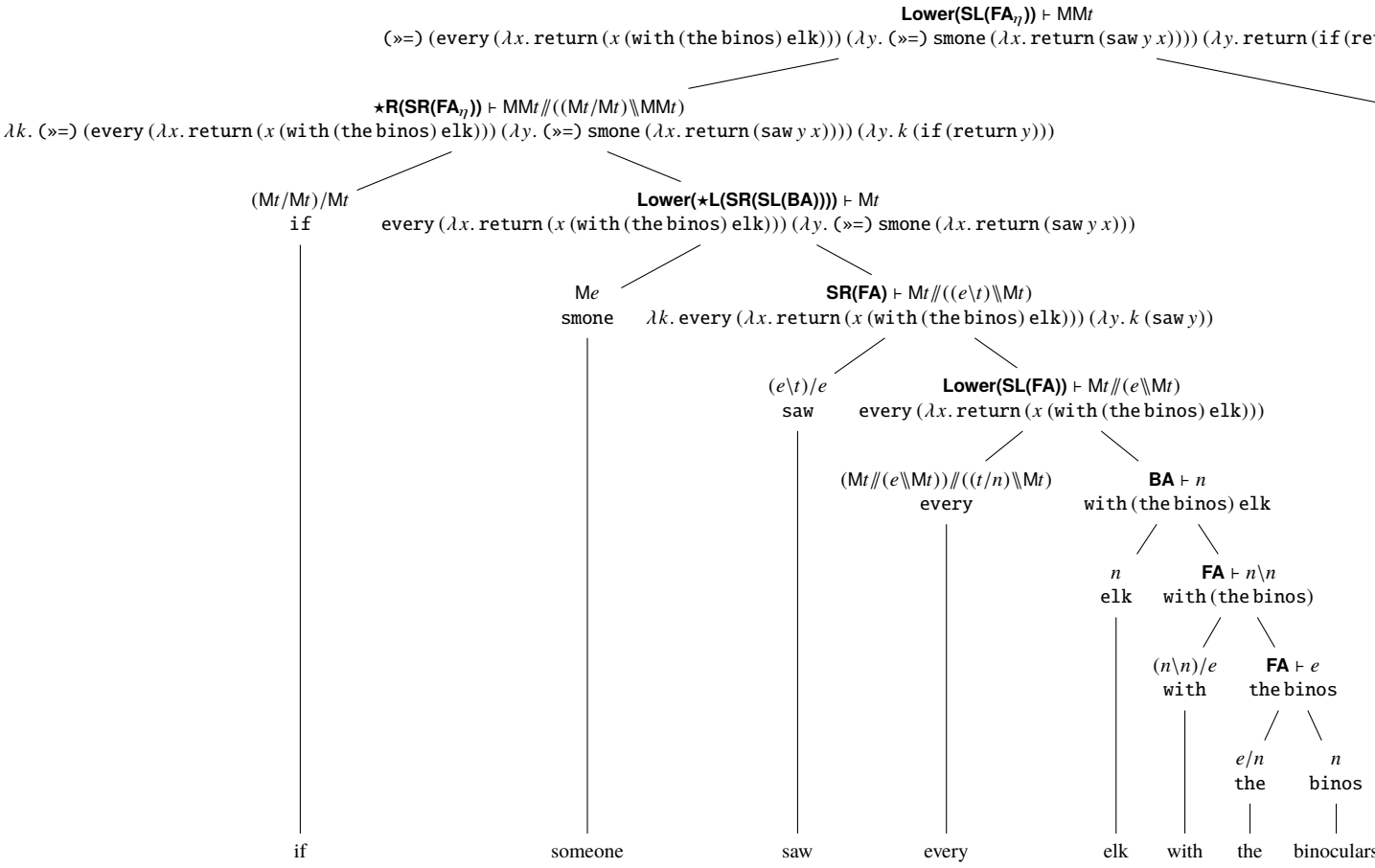

FA$_\eta$ ⊢ Mt
if ((»=) smone (λx. every (λx. return (x (with (the binos) elk))) (λy. return (saw y x)))) (return (left d))
FA ⊢ Mt/Mt
if ((»=) smone (λx. every (λx. return (x (with (the binos) elk))) (λy. return (saw y x))))
BA ⊢ t
left d
(Mt/Mt)/Mt
if
Lower(⋆L(SL(SR(BA)))) ⊢ Mt
(»=) smone (λx. every (λx. return (x (with (the binos) elk))) (λy. return (saw y x)))
e
d
Me
smone
SR(FA) ⊢ Mt⫽((e\t)\\M$t$)
λk. every (λx. return (x (with (the binos) elk))) (λy. k (saw y))
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt⫽(e\\M$t$)
every (λx. return (x (with (the binos) elk)))
(Mt⫽(e\\M$t$))⫽((t/n)\\M$t$)
every
BA ⊢ n
with (the binos) elk
n
elk
FA ⊢ n\n
with (the binos)
(n\n)/e
with
FA ⊢ e
the binos
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
dylan


Lower(SL(FA$_\eta$)) ⊢ Mt
(»=) ((»=) smone (λx. every (λx. return (x (with (the binos) elk))) (λy. return (saw y x)))) (λy. if (return
★R(SR(FA$_\eta$)) ⊢ Mt // ((Mt/Mt) \\ Mt)
λk. (»=) ((»=) smone (λx. every (λx. return (x (with (the binos) elk))) (λy. return (saw y x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(★L(SL(SR(BA)))) ⊢ Mt
(»=) smone (λx. every (λx. return (x (with (the binos) elk))) (λy. return (saw y x)))
Me
smone
SR(FA) ⊢ Mt // ((e\t) \\ Mt)
λk. every (λx. return (x (with (the binos) elk))) (λy. k (saw y))
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt // (e \\ Mt)
every (λx. return (x (with (the binos) elk)))
(Mt // (e \\ Mt)) // ((t/n) \\ Mt)
every
BA ⊢ n
with (the binos) elk
n
elk
FA ⊢ n\n
with (the binos)
(n\n)/e
with
FA ⊢ e
the binos
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars


Lower(SL(FA$_\eta$)) ⊢ MMt
(»=) ((»=) smone (λx. every (λx. return (x (with (the binos) elk))) (λy. return (saw y x)))) (λy. return (if (re
★R(SR(FA$_\eta$)) ⊢ MMt // ((Mt/Mt)\\MM$t$)
λk. (»=) ((»=) smone (λx. every (λx. return (x (with (the binos) elk))) (λy. return (saw y x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(★L(SL(SR(BA)))) ⊢ Mt
(»=) smone (λx. every (λx. return (x (with (the binos) elk))) (λy. return (saw y x)))
Me
smone
SR(FA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x (with (the binos) elk))) (λy. k (saw y))
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt // (e\\M$t$)
every (λx. return (x (with (the binos) elk)))
(Mt // (e\\M$t$)) // ((t/n)\\M$t$)
every
BA ⊢ n
with (the binos) elk
n
elk
FA ⊢ n\n
with (the binos)
(n\n)/e
with
FA ⊢ e
the binos
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars


FA$_\eta$ ⊢ Mt
if (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (return (left d))
FA ⊢ Mt/Mt
if (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x))))
BA ⊢ t
left d
(Mt/Mt)/Mt
if
Lower(⋆L(SR(SL(BA)))) ⊢ Mt
every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))
e
d
Me
smone
SR(FA) ⊢ Mt⫽((Mt\\e)\M$t$)
λk. every (λx. return (x (with (the binos) elk))) (λy. k (saw y))
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt⫽(e\\M$t$)
every (λx. return (x (with (the binos) elk)))
((Mt⫽(e\\M$t$))⫽((Mt/n)\\M$t$)
every
BA ⊢ n
with (the binos) elk
n
elk
FA ⊢ n\n
with (the binos)
(n\n)/e
with
FA ⊢ e
the binos
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars
dylan


Lower(SL(FA$_\eta$)) ⊢ Mt
(»=) (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (λy. if (return
★R(SR(FA$_\eta$)) ⊢ Mt /// ((Mt / Mt) \\ Mt)
λk. (»=) (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (λy. k (if (return y)))
(Mt / Mt) / Mt
if
Lower(★L(SR(SL(BA)))) ⊢ Mt
every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))
Me
smone
SR(FA) ⊢ Mt /// ((e\t) \\ Mt)
λk. every (λx. return (x (with (the binos) elk))) (λy. k (saw y))
(e\t) / e
saw
Lower(SL(FA)) ⊢ Mt /// (e \\ Mt)
every (λx. return (x (with (the binos) elk)))
(Mt /// (e \\ Mt)) /// ((t / n) \\ Mt)
every
BA ⊢ n
with (the binos) elk
n
elk
FA ⊢ n\n
with (the binos)
(n\n) / e
with
FA ⊢ e
the binos
e / n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars


Lower(SL(FA$_\eta$)) ⊢ MMt
(»=) (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (λy. return (if (re
★R(SR(FA$_\eta$)) ⊢ MMt // ((Mt/Mt)\\MM$t$)
λk. (»=) (every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))) (λy. k (if (return y)))
(Mt/Mt)/Mt
if
Lower(★L(SR(SL(BA)))) ⊢ Mt
every (λx. return (x (with (the binos) elk))) (λy. (»=) smone (λx. return (saw y x)))
Me
smone
SR(FA) ⊢ Mt // ((e\t)\\M$t$)
λk. every (λx. return (x (with (the binos) elk))) (λy. k (saw y))
(e\t)/e
saw
Lower(SL(FA)) ⊢ Mt // (e\\M$t$)
every (λx. return (x (with (the binos) elk)))
(Mt // (e\\M$t$)) // ((t/n)\\M$t$)
every
BA ⊢ n
with (the binos) elk
n
elk
FA ⊢ n\n
with (the binos)
(n\n)/e
with
FA ⊢ e
the binos
e/n
the
n
binos
if
someone
saw
every
elk
with
the
binoculars