objectraising

```
Quantifiers in Object Position, Raising and Type-Shifting

constants of type e : b g
variables of type t : p-r
constants of type <e,t> : P person hobbit wizard dwarf
constants of type <e*e,t> : love trust help
variables of type e : x-z
variables of type <et> : X Y
variables of type <e,<e,t>> : H
variables of type <<e,t>,t> : Q

multiple letter identifiers

use rule function application
use rule non-branching nodes
use rule lambda abstraction

define Boromir: b
define Gandalf: g

define person: Lx[person(x)]
define hobbit,a-hobbit: Lx[hobbit(x)]
define wizard,a-wizard: Lx[wizard(x)]
define dwarf,a-dwarf: Lx[dwarf(x)]

define loves: LyLx[love(x,y)]
define trusts,trust: LyLx[trust(x,y)]
define helps,help: LyLx[help(x,y)]

define is,are: LX.Lx[X(x)]

define everybody: LX[Ax[person(x) -> X(x)]]
define every,Every: LX[LY[Ax[X(x) -> Y(x)]]]
define some,Some: LX[LY[Ex[X(x) & Y(x)]]]

define raiseO: LH[LQ[Lx[Q(Ly[H(y)(x)])]]]

############################
# SECTION 1: SYNTACTIC RAISING
############################
```

```
exercise tree
title Quantifier Raising (object position)
directions Use function application and the lambda abstraction rule to evaluate these trees.

instructions Boromir loves everybody.
[.S [.DP everybody ] [.LP 1 [.S [.DP Boromir ] [.VP [.V loves ] [.DP t_1 ] ] ] ] ]

instructions Gandalf trusts every hobbit.
[.S [.DP Every [.NP hobbit ] ] [.LP 1 [.S [.DP Gandalf ] [.VP [.V trusts ] [.DP t_1 ] ] ] ] ]

instructions Some person loves everybody.
[.S [.DP everybody ] [.LP 1 [.S [.DP Some [.NP person ] ] [.VP [.V loves ] [.DP t_1 ] ] ] ] ]

instructions Every wizard trusts Boromir.
[.S [.DP Every [.NP wizard ] ] [.VP [.V trusts ] [.DP Boromir ] ] ]

instructions Some dwarf helps every wizard.
[.S [.DP Every [.NP wizard ] ] [.LP 1 [.S [.DP Some [.NP dwarf ] ] [.VP [.V helps ] [.DP t_1 ] ] ] ] ]

############################
# SECTION 2: TYPE SHIFTING
############################

exercise tree
title Object Raising (type shifting)
directions Here raiseO is an explicit node that lifts the verb so it can combine with a quantificational object.

instructions Boromir loves everybody.
[.S [.DP Boromir ] [.VP [.V' [.V raiseO ] [.V loves ] ] [.DP everybody ] ] ]

instructions Gandalf trusts every hobbit.
[.S [.DP Gandalf ] [.VP [.V' [.V raiseO ] [.V trusts ] ] [.DP Every [.NP hobbit ] ] ] ]

instructions Some person loves everybody.
[.S [.DP Some [.NP person ] ] [.VP [.V' [.V raiseO ] [.V loves ] ] [.DP everybody ] ] ]

instructions Every wizard loves everybody.
[.S [.DP Every [.NP wizard ] ] [.VP [.V' [.V raiseO ] [.V loves ] ] [.DP everybody ] ] ]

instructions Some dwarf helps every hobbit.
[.S [.DP Some [.NP dwarf ] ] [.VP [.V' [.V raiseO ] [.V helps ] ] [.DP Every [.NP hobbit ] ] ] ]
```