

predicatemodification

Adjectives, Predicate Modification, and Quantification

```
constants of type e : a b w g l m f s p
variables of type e : x y z
variables of type t : p q r
variables of type <et> : X Y
variables of type <<e,t>,<e,t>> : F G
variables of type <<e,t>,t> : Q
variables of type <e,<e,t>> : H
```

multiple letter identifiers

```
use rule function application
use rule non-branching nodes
use rule predicate modification
use rule lambda abstraction
```

```
define Aragorn: a
define Boromir: b
define Eowyn: w
define Gandalf: g
define Legolas: l
define Gimli: m
define Frodo: f
define Sam: s
define Pippin: p
```

```
define brave: Lx[brave(x)]
define tall: Lx[tall(x)]
define short: Lx[short(x)]
define elven: Lx[elven(x)]
define dwarven: Lx[dwarven(x)]
define gondorian: Lx[gondorian(x)]
```

```
define ranger: Lx[ranger(x)]
define king: Lx[king(x)]
define elf: Lx[elf(x)]
define dwarf: Lx[dwarf(x)]
define hobbit: Lx[hobbit(x)]
define captain: Lx[captain(x)]
define healer: Lx[healer(x)]
define archer: Lx[archer(x)]
define warrior: Lx[warrior(x)]
```

```

define traitor: Lx[traitor(x)]
define guard: Lx[guard(x)]

define experienced: LX.Lx[experienced(X,x)]
define alleged: LX.Lx[alleged(X,x)]
define former: LX.Lx[former(X,x)]
define fake: LX.Lx[fake(X,x)]
define potential: LX.Lx[potential(X,x)]

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

define is: LX[X]
define a: LX[X]
define an: LX[X]

define every,Every: LX[LY[Ax[X(x) -> Y(x)]]]
define some,Some: LX[LY[Ex[X(x) & Y(x)]]]
define no,No: LX[LY[~Ex[X(x) & Y(x)]]]

define and: Lp.[Lq.[p & q]]
define or: Lp.[Lq.[p V q]]

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

exercise tree
title Intersective Adjectives

instructions Aragorn is a brave ranger.
[.S [.DP Aragorn ] [.VP [.V is ] [.NP [.Det a ] [.N' [.AP brave ] [.N ranger ] ] ] ] ]

instructions Legolas is a tall elven archer.
[.S [.DP Legolas ] [.VP [.V is ] [.NP [.Det a ] [.N' [.AP tall ] [.N' [.AP elven ] [.N archer ] ] ] ] ] ]

instructions Gimli is a short brave dwarven warrior.
[.S [.DP Gimli ] [.VP [.V is ] [.NP [.Det a ] [.N' [.AP short ] [.N' [.AP brave ] [.N' [.AP dwarven ] [.N warrior ] ] ] ] ] ] ]

exercise tree
title Non-Intersective Adjectives

instructions Eowyn is an experienced healer.
[.S [.DP Eowyn ] [.VP [.V is ] [.NP [.Det an ] [.N' [.AP experienced ] [.N healer ] ] ] ] ]

instructions Boromir is a former Gondorian captain.

```

