morebasicexercises

```
Quantifiers, Connectives, and Prepositions

constants of type e : a-o s-w
variables of type t : p-r
constants of type <e,t> : P Q
constants of type <e*e,t> : R
variables of type e : x-z
variables of type <et> : X Y

define Sam: s
define Frodo: f
define Gimli: g
define Gandalf: d
define Sauron: n
define Mordor: m
define Gondor: o

define hobbit,a-hobbit: Lx.hobbit(x)
define elf,an-elf:      Lx.elf(x)
define dwarf,a-dwarf:   Lx.dwarf(x)
define brave:           Lx.brave(x)
define short:           Lx.short(x)
define runs,run:        Lx.runs(x)
define laughs,laugh:    Lx.laugh(x)
define trusts,trust:    LxLy.trusts(y,x)
define chases,chase:    LxLy.chases(y,x)

define in: LyLx.in(x,y)
define near:  LyLx.near(x,y)
define behind:LyLx.behind(x,y)

define is,are: LX[X]
define does: LX[X]

define not: LX.Lx.~X(x)
define it-is-not-the-case-that: Lp[~p]

define and:  Lp.[Lq.[p & q]]
define or:   Lp.[Lq.[p V q]]
define andP: LX.[LY.[Lx.[X(x) & Y(x)]]]
define orP:  LX.[LY.[Lx.[X(x) V Y(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)]]]

use rule function application

exercise tree
```

```
title Vacuous Items

[.S [.DP Sam ] [.VP is [.AP brave ] ] ]
[.S [.DP Sam ] [.VP does [.V laugh ] ] ]
[.S [.DP Sam ] [.VP does [.NegP not [.V laugh ] ] ] ]
[.NegP it-is-not-the-case-that [.S [.DP Sam ] [.VP does [.V laugh ] ] ] ]

exercise tree
title Predicate vs. Sentential Negation

[.S [.DP Sam ] [.VP is [.NegP not [.AP brave ] ] ] ]
[.NegP it-is-not-the-case-that [.S [.DP Sam ] [.VP is [.AP brave ] ] ] ]

exercise tree
title Connectives

[.S [.S [.DP Sam ] [.VP runs ] ] [.ConjP and [.S [.DP Frodo ] [.VP runs ] ] ] ]
[.S [.S [.DP Sam ] [.VP runs ] ] [.ConjP or [.S [.DP Frodo ] [.VP runs ] ] ] ]
[.S [.DP Gimli ] [.VP is [.AP [.AP short ] [.ConjP andP [.AP brave ] ] ] ] ]
[.S [.DP Frodo ] [.VP is [.AP [.AP brave ] [.ConjP orP [.AP short ] ] ] ] ]

exercise tree
title Prepositions as Relational Predicates

[.S [.DP Gandalf ] [.VP is [.PP in [.DP Mordor ] ] ] ]
[.NegP it-is-not-the-case-that [.S [.DP Gandalf ] [.VP is [.PP in [.DP Mordor ] ] ] ] ]
[.S [.S [.DP Sam ] [.VP is [.PP in [.DP Mordor ] ] ] ] [.ConjP and [.S [.DP Frodo ] [.VP is [.PP in [.DP Mordor ] ] ] ] ] ]
[.S [.DP Gimli ] [.VP is [.PP near [.DP Mordor ] ] ] ]
[.S [.DP Gandalf ] [.VP is [.PP behind [.DP Sauron ] ] ] ]
[.S [.S [.DP Gandalf ] [.VP is [.PP behind [.DP Sauron ] ] ] ] [.ConjP and [.S [.DP Gimli ] [.VP is [.PP near [.DP Mordor ] ] ] ] ] ]
[.NegP it-is-not-the-case-that [.S [.S [.DP Sam ] [.VP is [.PP in [.DP Mordor ] ] ] ] [.ConjP or [.S [.DP Frodo ] [.VP is [.PP in [.DP Mordor ] ] ] ] ] ] ]

exercise tree
title Mixed Practice

[.S [.S [.DP Sam ] [.VP does [.V run ] ] ] [.ConjP and [.S [.DP Frodo ] [.VP laughs ] ] ] ]
[.S [.S [.DP Sam ] [.VP is [.NegP not [.AP brave ] ] ] ] [.ConjP or [.S [.DP Frodo ] [.VP is [.AP short ] ] ] ] ]
[.S [.S [.DP Gandalf ] [.VP is [.PP in [.DP Mordor ] ] ] ] [.ConjP and [.S [.DP Gimli ] [.VP is [.AP brave ] ] ] ] ]
[.S [.DP Sam ] [.VP is [.AP [.AP brave ] [.ConjP andP [.AP short ] ] ] ] ]
[.S [.DP Gandalf ] [.VP is [.AP [.AP brave ] [.ConjP andP [.PP in [.DP Mordor ] ] ] ] ] ]
[.S [.DP Every [.NP hobbit ] ] [.VP [.V trusts ] [.DP Gandalf ] ] ]
[.S [.DP Some [.NP dwarf ] ] [.VP [.V chases ] [.DP Sauron ] ] ]
[.S [.DP No [.NP elf ] ] [.VP [.V trusts ] [.DP Sauron ] ] ]
[.S [.S [.DP Every [.NP hobbit ] ] [.VP [.V trusts ] [.DP Gandalf ] ] ] [.ConjP and [.S [.DP Some [.NP dwarf ] ] [.VP [.V chases ] [.DP Sauron ] ] ] ] ]
[.S [.NegP it-is-not-the-case-that [.S [.DP No [.NP elf ] ] [.VP [.V trusts ] [.DP Sauron ] ] ] ] [.ConjP and [.S [.DP Sam ] [.VP is [.AP brave ] ] ] ] ]
[.S [.S [.DP Some [.NP hobbit ] ] [.VP [.V trusts ] [.DP Frodo ] ] ] [.ConjP or [.S [.DP Gandalf ] [.VP is [.PP in [.DP Gondor ] ] ] ] ] ]
[.S [.S [.DP Every [.NP dwarf ] ] [.VP is [.AP [.AP short ] [.ConjP andP [.AP brave ] ] ] ] ] [.ConjP and [.S [.DP Gandalf ] [.VP is [.PP behind [.DP Sauron ] ] ] ] ] ]
```