Lambda calculus syntax reference

Simply-typed lambda calculus ($\lambda^{\to}$)

N.B. A,B stand for types, and L,M,N stand for terms.

Basic types

$\mathcal{B}$ .... base type (p5)
A→B .... function type (argument A, result B) (p5)

Basic terms

x .... variable (p6)
λx:A.M .... function (parameter x, parameter type A, body M) (p7)
M N .... function application (function M, body N) (p7)

Extra types

A×B .... type of products (pairs) of A and B (p9)
A+B .... type of sums of A or B (p10)

Extra terms

⟨M,N⟩ .... build a pair from M and N (p9)
fst M .... 1st projection: extract the 1st component of a pair (p9)
snd M .... 2nd projection: extract the 2nd component of a pair (p9)

inl M .... left injection into a sum (p10)
inr M .... right injection into a sum (p10)
case L **of** x.M | y.N .... reduce to M or N if L is **inl** x or **inr** y (p11)

System F

(Everything from $\lambda^{\to}$, plus the following)

Basic types

$\forall\alpha :: K.A$ .... universal type: for all α of kind K, A (p12)

Basic terms

$\Lambda\alpha :: K.M$ .... a function that takes a type α and returns a term M (p13)
M [A] .... application of the function M to the type A (p13)

Extra types

$\exists\alpha :: K.A$ .... existential type: for some α of kind K, A (p14)

Extra terms

pack B,M **as** $\exists\alpha :: K.A$ ..... pack together B (a type) and M (a term) (p14)
open M **as** α,x **in** M' .. unpack M, binding α (a type) and x (a term) (p14)