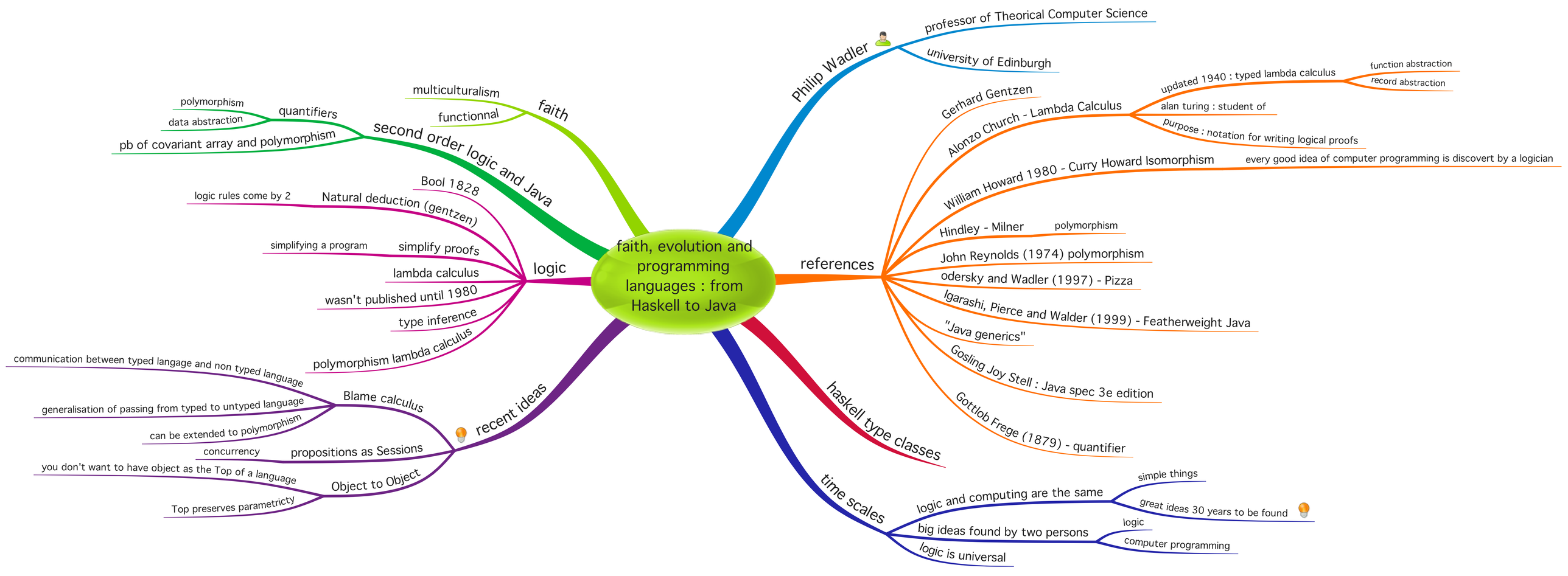
faith, evolution and programming languages : from Haskell to Java
Philip Wadler
professor of Theorical Computer Science
university of Edinburgh
references
Gerhard Gentzen
Alonzo Church - Lambda Calculus
updated 1940 : typed lambda calculus
function abstraction
record abstraction
alan turing : student of
purpose : notation for writing logical proofs
William Howard 1980 - Curry Howard Isomorphism
every good idea of computer programming is discovert by a logician
Hindley - Milner
polymorphism
John Reynolds (1974) polymorphism
odersky and Wadler (1997) - Pizza
Igarashi, Pierce and Walder (1999) - Featherweight Java
"Java generics"
Gosling Joy Stell : Java spec 3e edition
Gottlob Frege (1879) - quantifier
haskell type classes
time scales
logic and computing are the same
simple things
great ideas 30 years to be found
big ideas found by two persons
logic
computer programming
logic is universal
faith
multiculturalism
functionnal
second order logic and Java
quantifiers
polymorphism
data abstraction
pb of covariant array and polymorphism
logic
Bool 1828
Natural deduction (gentzen)
logic rules come by 2
simplify proofs
simplifying a program
lambda calculus
wasn't published until 1980
type inference
polymorphism lambda calculus
recent ideas
Blame calculus
communication between typed langage and non typed language
generalisation of passing from typed to untyped language
can be extended to polymorphism
propositions as Sessions
concurrency
Object to Object
you don't want to have object as the Top of a language
Top preserves parametricty