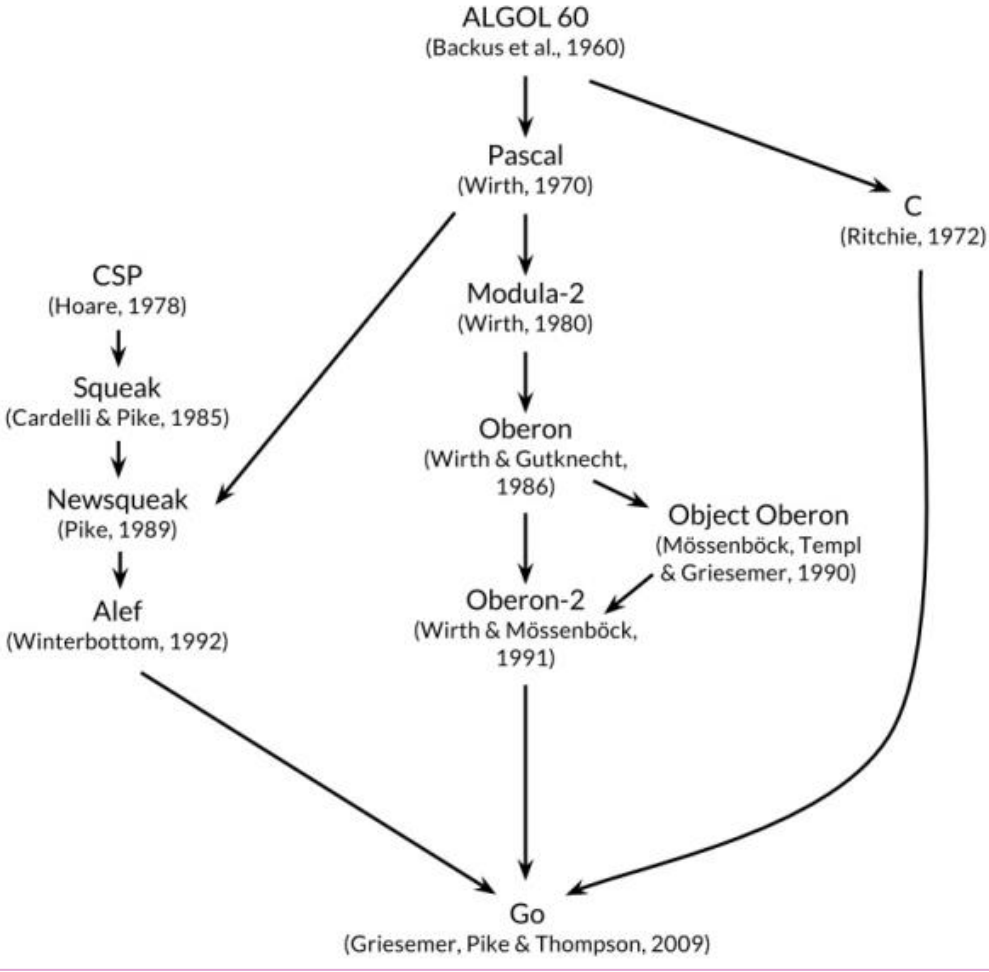


go基础

Go 家族树



基本类型

- 整数
 - 有符号整数: [int8、int16、int32 和 int64](#)
 - 无符号整数: [uint8、uint16、uint32 和 uint64](#)
 - 对应特定 CPU 平台机器字大小的有符号整数: [int\(32 或 64 bit\)](#)
 - 对应特定 CPU 平台机器字大小的无符号整数: [uint\(32 或 64 bit\)](#)
 - 无符号的整数类型 [uintptr](#)，没有指定具体的 bit 大小但是足以容纳指针，在底层编程时才需要
- 浮点数
 - [float32](#)
 - [float64](#)
- 复数
 - [complex64](#)
 - [complex128](#)
- 布尔型
 - [true](#)
 - [false](#)
- 字符串
 - 子文本字符串通常被解释为采用 UTF8 编码的 Unicode 码点 (rune) 序列，第 i 个字节并不一定是字符串的第 i 个字符，因为对于非 ASCII 字符的 UTF8 编码会要两个或多个字节。主题
- 原生的字符串
 - 一个原生的字符串面值形式是 `...``，使用反引号代替双引号。
 - 在原生的字符串面值中，没有转义操作：全部的内容都是字面的意思，包含退格和换行，因此一个程序中的原生字符串面值可能跨越多行。
- 常量
 - 常量表达式的值在编译期计算，而不是在运行期。每种常量的潜在类型都是基础类型：[boolean、string 或数字](#)。
 - [const 不能用于初始化 struct 对象。](#)
- 无类型常量
 - [无类型的布尔型](#)
 - [无类型的整数](#)
 - [无类型的字符](#)
 - [无类型的浮点数](#)
 - [无类型的复数](#)
 - [无类型的字符串](#)

复合数据类型

- 数组
 - 数组是一个由固定长度的特定类型元素组成的序列，一个数组可以由零个或多个元素组成。
- slice
 - 指针,指向第一个 slice 元素对应的底层数组元素的地址,要注意 slice 的第一个元素并不一定是数组的第一个元素。
 - Slice (切片) 代表变长的序列，序列中每个元素都有相同的类型。
 - 长度,对应 slice 中元素的数目,长度不能超过容量,内置函数 [len](#) 返回 slice 长度。
 - 容量,容量一般是从 slice 的开始位置到底层数据的结尾位置，[cap](#) 返回 slice 容量。
- Map
 - Map 的迭代顺序是不确定的，并且不同的哈希函数实现可能导致不同的遍历顺序。
- 结构体
 - 一个命名为 S 的结构体类型将不能再包含 S 类型的成员：因为一个聚合的值不能包含它自身。（该限制同样适用于数组。）

运算

- 以下运算符按照优先级递减
 - [*, /, %, <<, >>, &, &^](#)
 - [+, -, |, ^](#)
 - [==, !=, <, <=, >, >=](#)
 - [&&](#)
 - [||](#)