

第2章 C运算符和表达式

——算术表达式中的自动类型转换

本节要讨论的主要问题

- 不同类型的数据进行运算，其结果是什么类型？
- 类型提升的规则是什么？为什么要这样设计类型提升规则？



自动类型转换

■ 算术表达式中

- * 问题：相同类型数据的运算结果的类型是什么？
- * 还是该类型
- * 例如，整数除法

$$11 / 5 = 2$$



自动类型转换

■ 算术表达式中

- * 问题：不同类型数据的运算结果的类型是什么？
- * 取值范围较大的那种类型
 - * 例如，浮点数除法

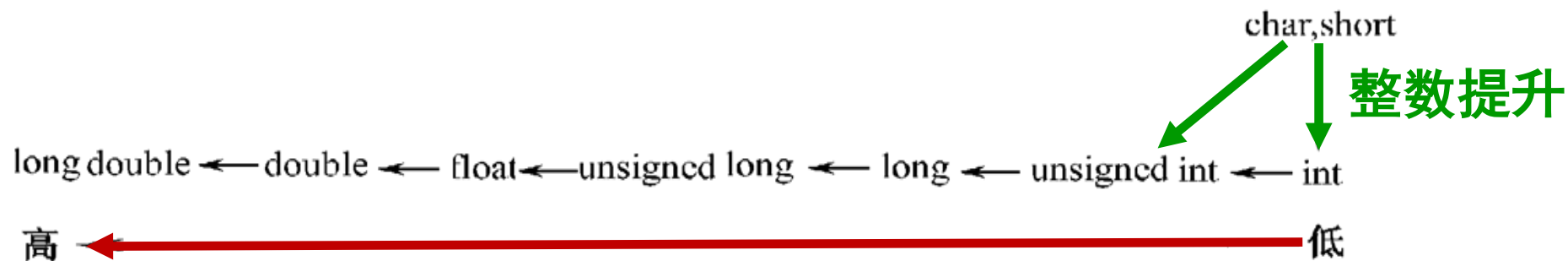
$$11.0 / 5 = 11.0 / 5.0 = 2.2$$

- * C编译器将所有操作数都转换成取值范围较大的操作数的类型
——类型提升 (Type Promotion)



自动类型转换

- 问题：类型提升的规则是什么？
 - * C99增强了整数提升规则
 - * char和short都可以直接提升为unsigned int型



根据参与运算的操作数类型
从低级别向高级别自动转换

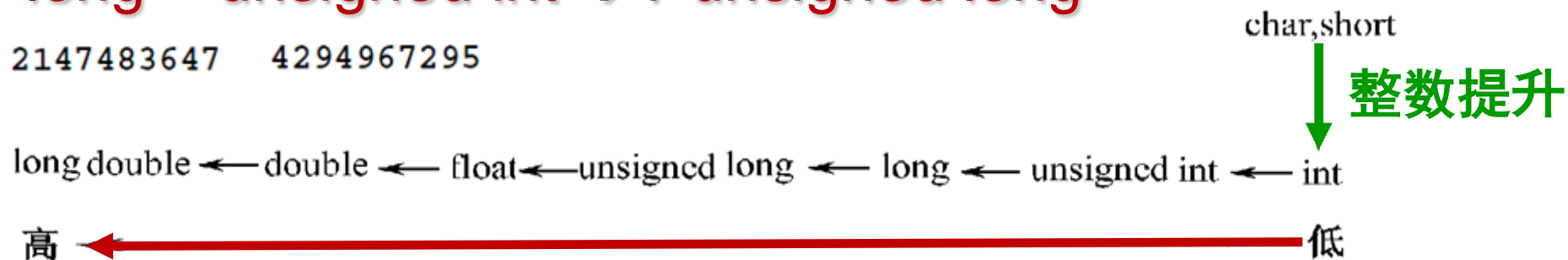
自动类型转换

- 问题：类型提升的规则是什么？
 - * 如何理解这里的横向箭头和纵向箭头？

`int + float → float + float`

`long + unsigned int → ? unsigned long`

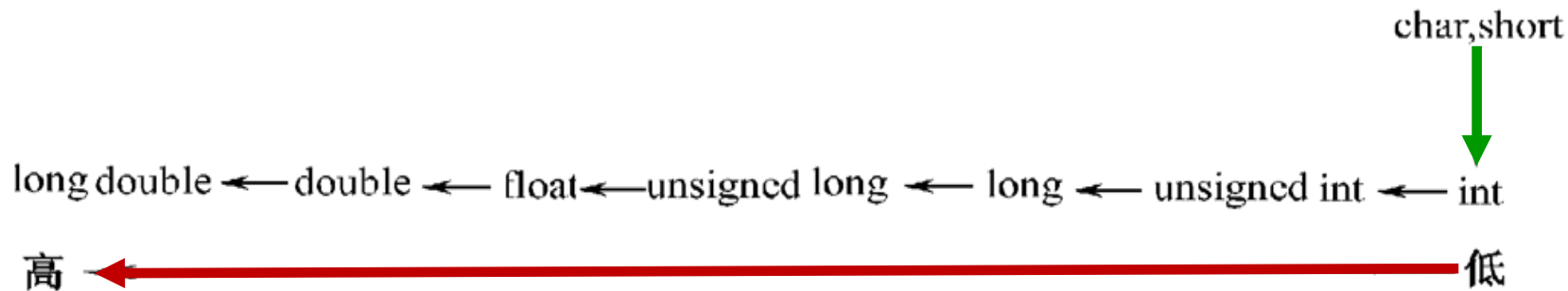
2147483647 4294967295



根据参与运算的操作数类型
从低级别向高级别自动转换

自动类型转换

■ 为什么这样设计类型提升规则？



根据参与运算的操作数类型
从低级别向高级别自动转换

