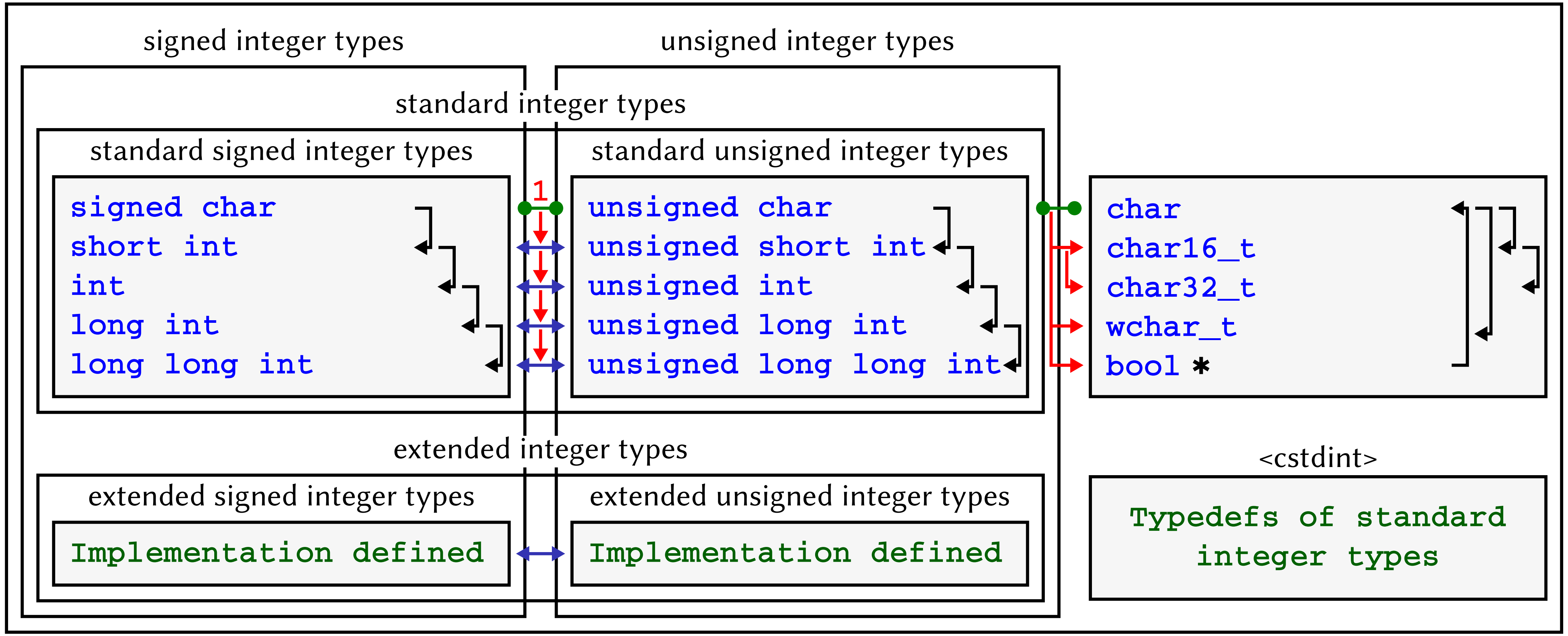
integral (or integer) types



signed integer types

unsigned integer types

standard integer types

standard signed integer types

signed char
short int
int
long int
long long int

1

standard unsigned integer types

unsigned char
unsigned short int
unsigned int
unsigned long int
unsigned long long int

char
char16_t
char32_t
wchar_t
bool *

extended integer types

extended signed integer types

Implementation defined

extended unsigned integer types

Implementation defined

<cstdint>

Typedefs of standard integer types

T1 ●–● T2 Narrow character types, same amount of storage with sizeof(T) == 1 byte , same alignment requirements, same object representation, and same integer conversion rank.

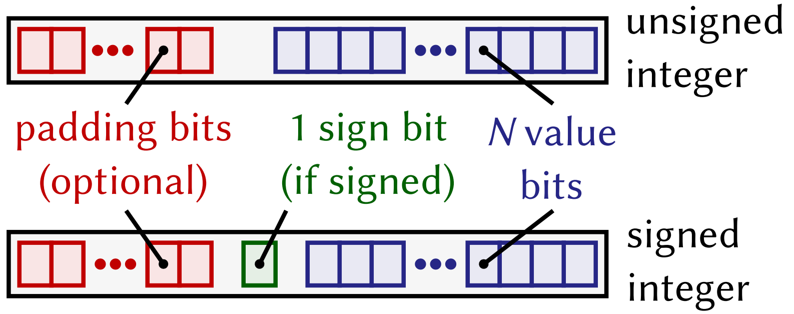
T1 ↔ T2 Corresponding signed/unsigned integer types, same amount of storage sizeof(T1) == sizeof(T2), same alignment requirements, same object representation, and same integer conversion rank.

↓ T1 T2 sizeof(T2) is greater than or equal to sizeof(T1).

T1 ┐ T2 ↲ The integer conversion rank of T2 is greater than the integer conversion rank of T1.

* bool values can be false or true, and they can be promoted to int values with false becoming 0 and true becoming 1.

unsigned/signed integer representation



•The object representation of integer types includes optional padding bits, one sign bit for signed types equals to zero for positive values, and N value bits given by std::numeric_limits::digits. The bit ordering is implementation defined.

•The value representation of integral types uses a pure binary numeration system. Unsigned integers arithmetic is modulo 2^N.

•The range of non-negative values of a signed integer type is a subrange of the corresponding unsigned integer type. Value representation of each corresponding signed/unsigned type is the same.

•Narrow character types do not have padding bits. Each possible bit pattern of unsigned narrow character types represents a distinct number.

•∀ unsigned char i∈⟦0, 255⟧, ∃ char j, static_cast<char>(i) == j && static_cast<decltype(i)>(j) == i .

•A prvalue of an integral type T1 is can be converted to a prvalue of another integer type T2. If T2 is unsigned, the resulting value is the least unsigned integer congruent to the source integer, modulo 2^N. If T2 is signed, the value is unchanged if it can be represented in T2; otherwise, the value is implementation defined.