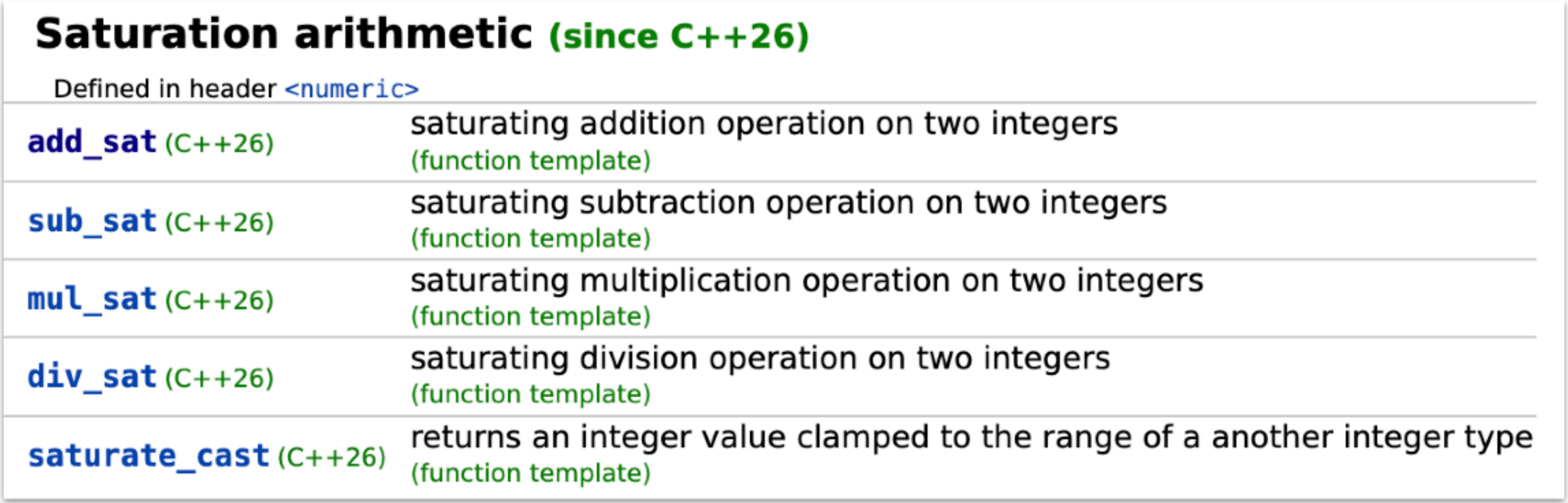
C++23/26*

- **Bounds:**
  - Use the ***flux*** library (index based ranges) or `std::ranges`
  - Use hardened std library (`_LIBCPP_HARDENING_MODE_DEBUG/FAST=1`)
- **Lifetime:**
  - Static analyser
- **Initialisation:**
  - Static analyser
- **Arithmetic:**
  - Saturating numeric operations (C++26)
  - Use ***`-ftrapv`*** to generate traps for signed integer overflow
- **Thread:** Tsan

34



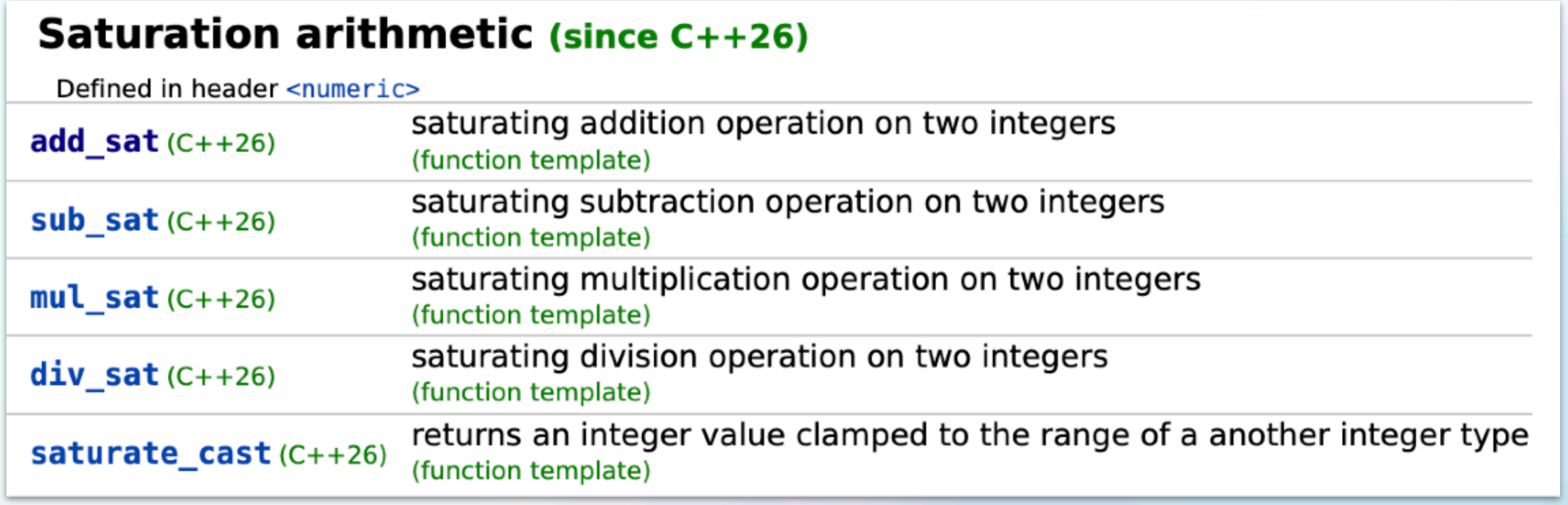
Saturation arithmetic (since C++26)

Defined in header `<numeric>`

add_sat (C++26)	saturating addition operation on two integers (function template)
sub_sat (C++26)	saturating subtraction operation on two integers (function template)
mul_sat (C++26)	saturating multiplication operation on two integers (function template)
div_sat (C++26)	saturating division operation on two integers (function template)
saturate_cast (C++26)	returns an integer value clamped to the range of a another integer type (function template)

C++23/26*

- **Bounds:**
  - Use the ***flux*** library (index based ranges) or `std::ranges`
  - Use hardened std library (`_LIBCPP_HARDENING_MODE_DEBUG/FAST=1`)
- **Lifetime:**
  - Static analyser
- **Initialisation:**
  - Static analyser
- **Arithmetic:**
  - Saturating numeric operations (C++26)
  - Use ***`–ftrapv`*** to generate traps for signed integer overflow
- **Thread:** Tsan



Saturation arithmetic (since C++26)

Defined in header `<numeric>`

`add_sat` (C++26)	saturating addition operation on two integers (function template)
`sub_sat` (C++26)	saturating subtraction operation on two integers (function template)
`mul_sat` (C++26)	saturating multiplication operation on two integers (function template)
`div_sat` (C++26)	saturating division operation on two integers (function template)
`saturate_cast` (C++26)	returns an integer value clamped to the range of a another integer type (function template)

Bounds/Lifetime:
Flux - Sequence Based Programming