

# 關於C++模板的類型轉換的討論



## 目錄

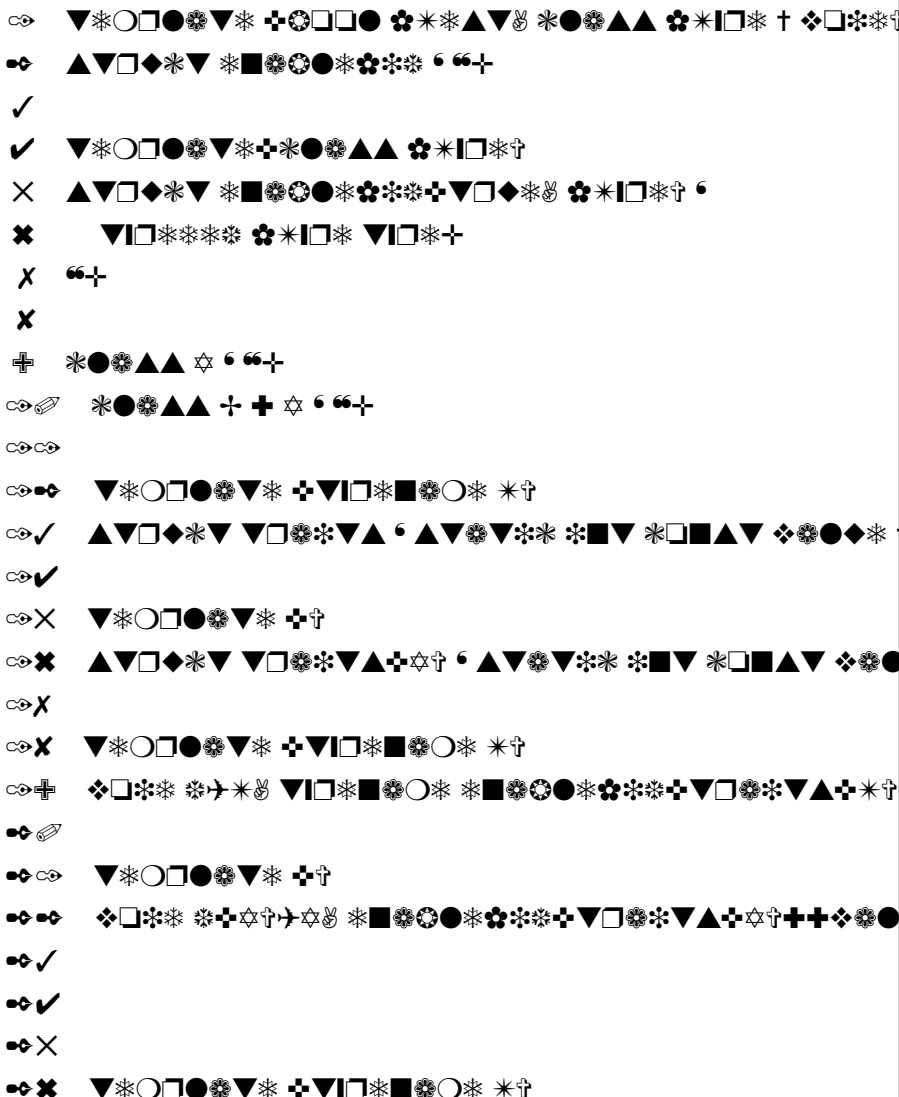
- [討論地址](#)
- [原問題](#)
- [我的解答](#)
  - [首先看ff的情況。](#)
  - [再來看f的情況。](#)

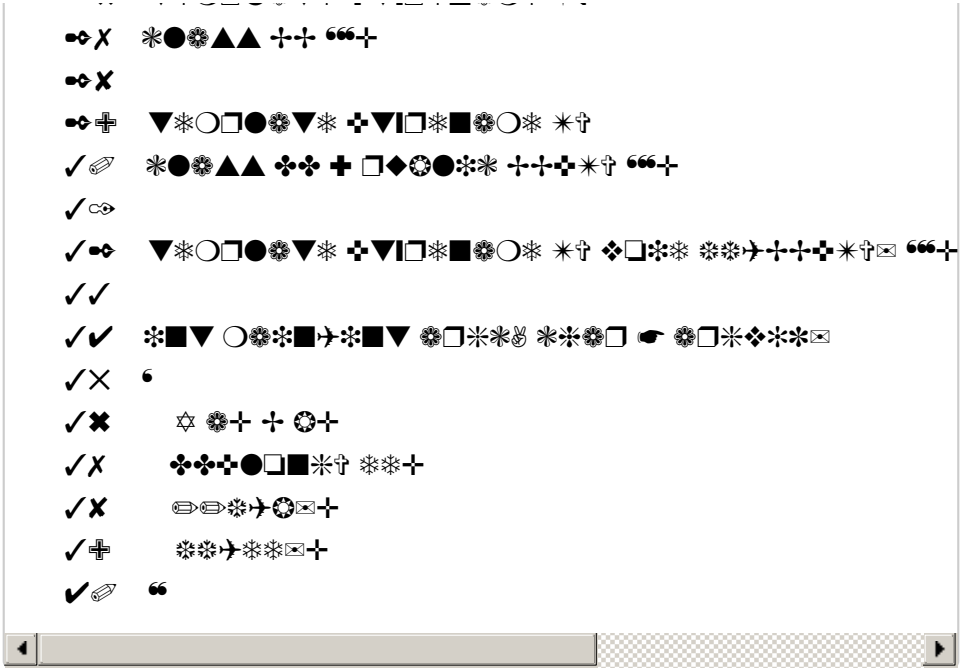
這兩天在飲水思源的C板，關於C++模板的類型轉換的一個討論，後面是我的解答。

## [討論地址](#)

# 原問題

今天在書上看到模板演繹的時候可以允許cast-down，於是我寫了個東西：





奇怪的是重載決議的時候，`*` 的情況下它就不讓我特化的 `*+*` 進來。

但是在 `**` 的情況下，`**+*+*+*+*+*` 卻進來了。

在VC10和GCC3.4下測試

# 我的解答

我們來設身處地地作為編譯器，看一遍到底發生了什麼。

約定符號 `~`：`~+~` 是把 `+` 帶入 `*+*` 的參數 `*` 之後實例化得到的結果。

## 首先看ff的情況。

[illegible]

👁️ ❄️ ✈️ ❄️ 📺 ➕

這句，首先計算重載函數集合。

第一步,需要從參數  $\clubsuit\spadesuit\heartsuit\blacklozenge\blacksquare\blacksuit\clubsuit\spadesuit\heartsuit\blacklozenge\blacksquare\blacksuit$  推斷  $\clubsuit\clubsuit\heartsuit\spadesuit$  的  $\spadesuit$ 。根據函數模板參數推斷規則：

+ \* □ ✖ ✖ + ✖ ✖ ● ● ▲ ▲ ▲ ✖ ▼ ○ □ ● ● ▼ ▲ ✖ ● ○ ✖ + ✖ 寧 ✖ 類型的參

於是編譯器推斷 \* 為   \* 。這裏就算不是 ++ 而是完全無關的 ++ 都可以推斷成功，只要 ++ 也是一個 ++\* 形式的模板。

第二步，模板特化匹配。因為只有一個模板，所以匹配了最泛化的  $\text{int}^*$ 。

第三步，模板實例化。

推斷了  $\bullet \square \blacksquare \ast$   $\rightarrow \rightarrow \ast$  之後，編譯器實例化  $\ast \ast \ast \rightarrow \bullet \square \blacksquare \ast$ 。

重載函數集合： “❄️✂️●◻️■❄️”

然後重載抉擇找到唯一的可匹配的實例 `int * f(int *, int *, int *)`，檢查實際參數 `int a, b, c` 可以隱式轉換到 形式參數 `int * p, int * q, int * r`，從而生成了這次函數調用。

再來看f的情況。

計算候選重載函數集合。

第一步，對所有  $\ast$  模板推斷實參。根據函數模板參數推斷規則：

帶有  $\text{+*}\square\text{**}\text{+}\text{**}\text{+}$  類型的參數，可以用於推斷  $\text{+*}\square\text{**}\text{+}\text{**}\text{+}$

於是  $\div$    $\div$   $*$  被推斷出來了。

## 第二步，模板特化匹配。

這裏  $\oplus$  不是  $\odot$  , 所以不能用  $\otimes \oplus \odot$  特化, 只能用  $\otimes \oplus \ast$  模板。

### 第三步，模板實例化。

✦ 帶 入 ✨✧✨ 實 例 化 成 ✨✂✦ 的 過 程 中，實 例 化  
▼□✨▼▲✂✦。

由於沒有針對  $\oplus$  的特化，所以用  $\nabla \square \otimes \nabla \blacktriangle \oplus * \dagger$  模板，  
 $\nabla \square \otimes \nabla \blacktriangle \div \dagger \oplus \oplus \diamond \bullet \blacklozenge * \dagger \otimes \bullet \bullet \blacktriangle *$ ，進而  
 $* \blacksquare \otimes \bullet \bullet \bullet \bullet \oplus * \otimes \div \otimes \bullet \bullet \blacklozenge \blacktriangle *$  沒有  $\nabla \square *$ ，出錯。

唯一的模板匹配出錯，重載函數集合爲空，SFINAE原則不能找到合適的匹配，於是報錯。