## 1.数据流分析总览

may analysis：输出可能正确的信息（需做over-approximation优化，才能成为Safe-approximation安全的近似，可以有误报-completeness），注意大多数静态分析都是may analysis

must analysis：输出必须正确的信息（需做under-approximation优化，才能成为Safe-approximation安全的近似，可以有漏报-soundness）

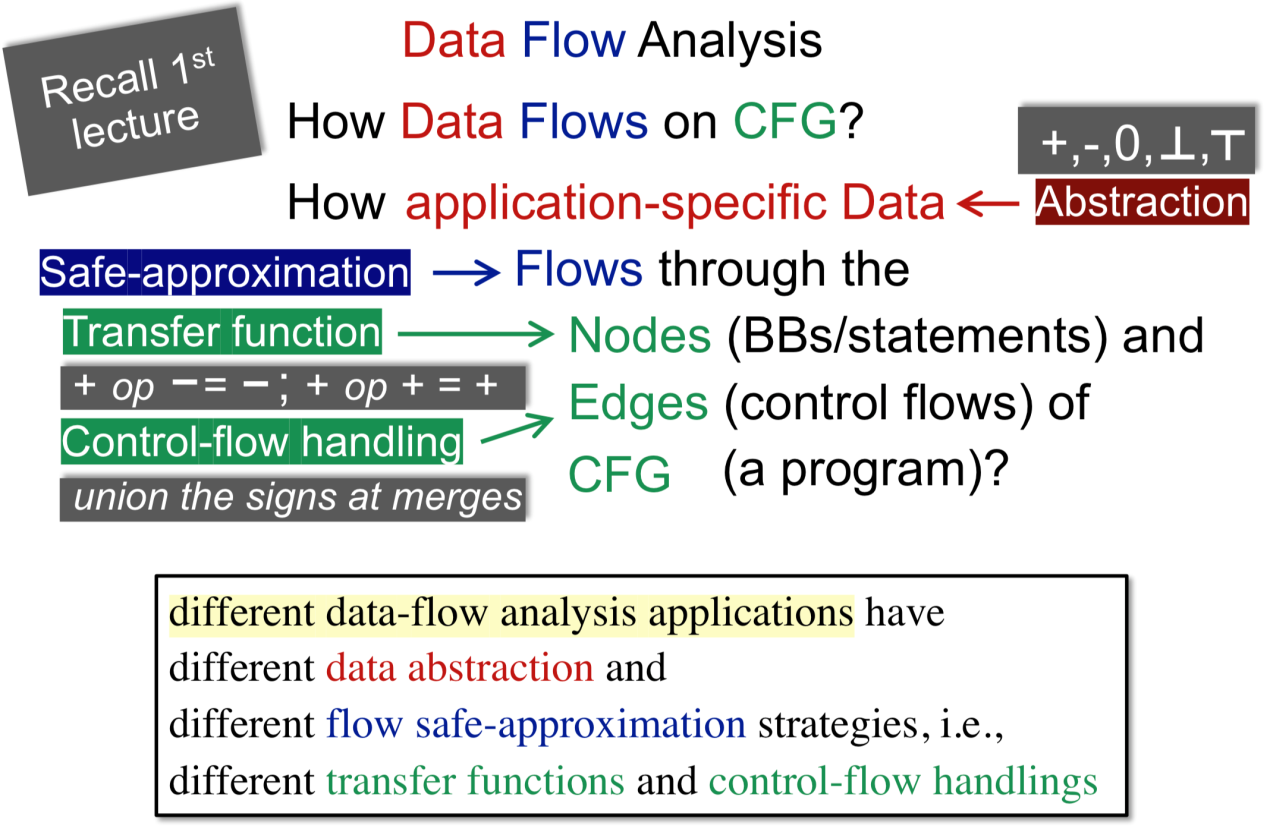
Nodes (BBs/statements)、Edges (control flows)、CFG (a program)

**例如**：

application-specific Data <- abstraction (+/-/0)

Nodes <- Transfer function

Edges <- Control-flow handling

不同的数据流分析 有 不同的数据**抽象表达** 和 不同的**安全近似策略**，如 不同的 **转换规则** 和 **控制流**处理。

## 2.预备知识

**输入/输出状态**：程序执行前/执行后的状态（本质就是抽象表达的数据的状态，如变量的状态）。

**数据流分析的结果**：最终得到，每一个程序点对应一个数据流值(data-flow value)，表示该点所有可能程序状态的一个抽象。例如，我只关心x、y的值，我就用抽象来表示x、y所有可能的值的集合（输入/输出的值域/约束），就代表了该程序点的程序状态。

Forward Analysis前向分析：按程序执行顺序的分析。OUT[s]=fs(IN[s])，s-statement

Backward Analysis反向分析：逆向分析。IN[s]=fs(OUT[s])

**控制流约束**：约束求解做的事情，推断计算输入到输出，或反向分析。

