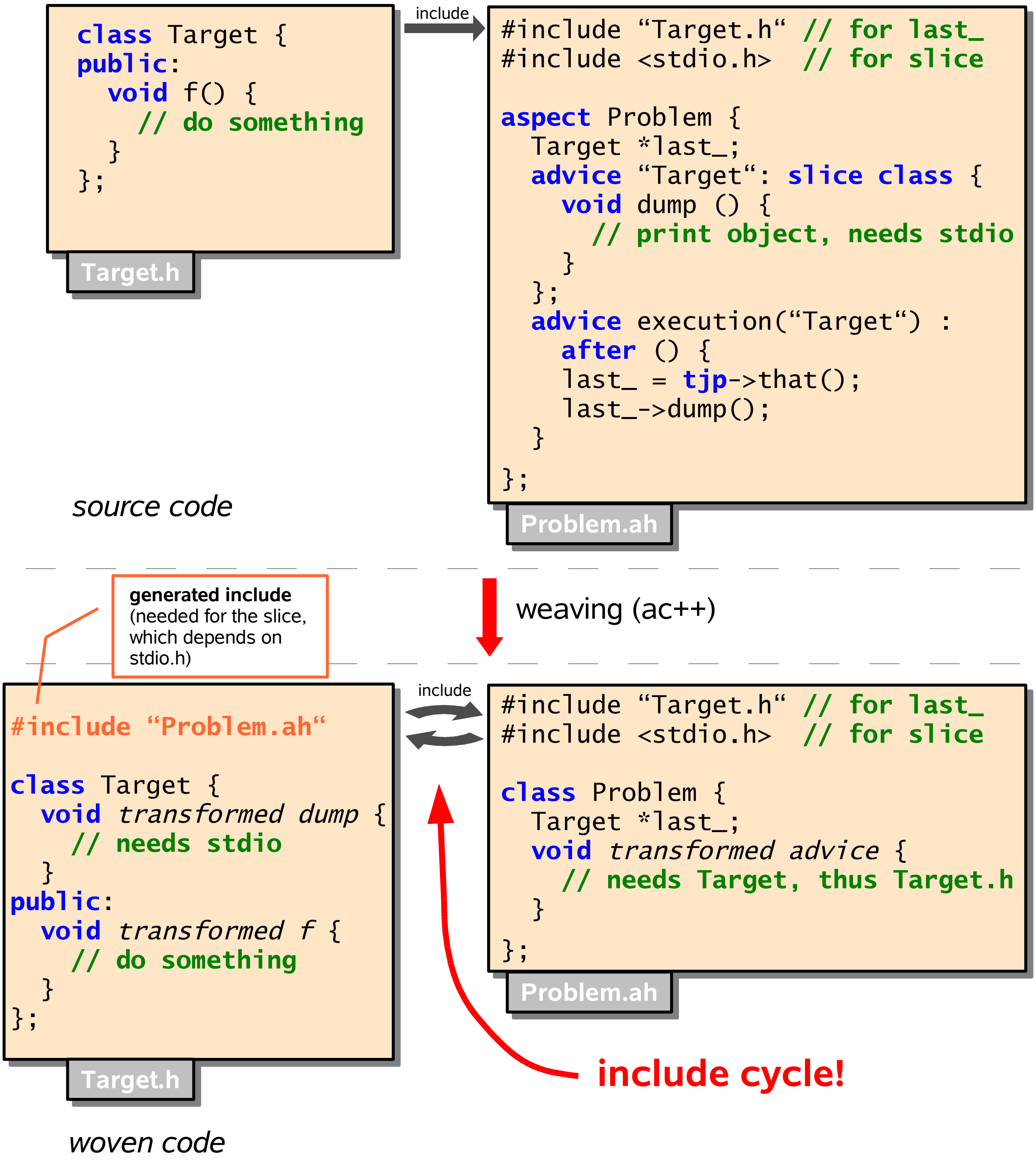

```
class Target {
public:
   void f() {
      // do something
   }
};
```

Target.h

include

```
#include “Target.h“ // for last_
#include <stdio.h>   // for slice

aspect Problem {
   Target *last_;
   advice “Target“: slice class {
      void dump () {
         // print object, needs stdio
      }
   };
   advice execution(“Target“) :
      after () {
      last_ = tjp->that();
      last_->dump();
   }
};
```

Problem.ah

source code

weaving (ac++)

generated include (needed for the slice, which depends on stdio.h)

```
#include “Problem.ah“

class Target {
   void transformed dump {
      // needs stdio
   }
public:
   void transformed f {
      // do something
   }
};
```

Target.h

include

```
#include “Target.h“ // for last_
#include <stdio.h>   // for slice

class Problem {
   Target *last_;
   void transformed advice {
      // needs Target, thus Target.h
   }
};
```

Problem.ah

include cycle!

woven code