清華大學

同步数据流语言可信编译器的设计

2014-06-14

同步数据流语言可信编译器的设计

清華大學

- 可信编译器研究
- 同步数据流语言可信编译器研究
- Lustre* 到 C 的可信编译器 L2C
- L2C 项目进展情况
- 总结及展望

清華大學

✧ 为什么要形式化验证代码生成器/编译器

– 误编译：从正确的源程序可能产生错误的目标代码

例　Csmith 工具测试 C 编译器

Abstract. Compilers should be correct. To improve the quality of C compilers, we created Csmith, a randomized test-case generation tool, and spent three years using it to find compiler bugs. During this period we reported more than 325 previously unknown bugs to compiler developers. Every compiler we tested was found to crash and also to silently generate wrong code when presented with valid input.

（摘自 PLDI 2011 论文）

可信编译器研究

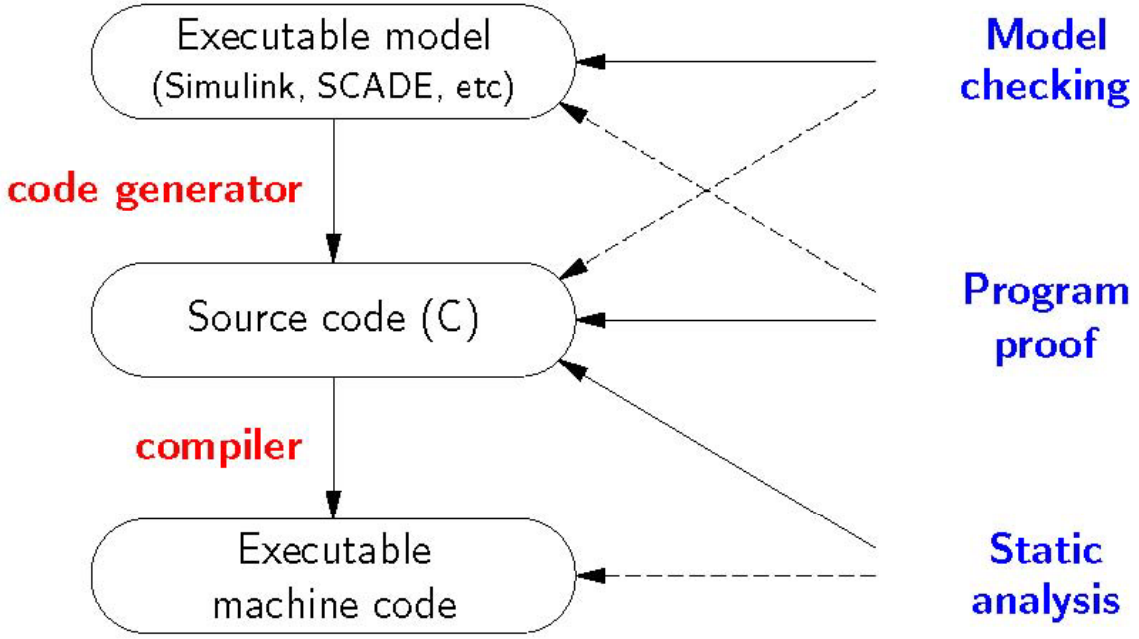
清華大學

✧ 为什么要形式化验证代码生成器/编译器

– 误编译的危害

关键系统的测试不能不考虑可能由编译器引入的错误

化很大精力在源程序级的验证工作在目标程序及失效



（图片源于
X.Leroy
CGO 2011
特邀报告）

清華大學

✧ 为什么要形式化验证代码生成器/编译器

– 传统的方法能否彻底解决误编译问题？

较新版 GCC(4.7) 的测试集包含 2853 个源程序用例

商用的 PlumHall C 语言标准符合测试集有 29424 个用例

Bug-hunting 工具（如 Csmith）可能产生更多的用例

清華大學

✧ 为什么要形式化验证代码生成器/编译器

– 形式化验证过的编译器有很好的口碑

例　Csmith 工具测试 CompCert 编译器

The striking thing about our CompCert results is that the middle end bugs we found in all other compilers are absent. As of early 2011, the under-development version of CompCert is the only compiler we have tested for which Csmith cannot find wrong-code errors. This is not for lack of trying: we have devoted about six CPU-years to the task.

（摘自 PLDI 2011 论文）

可信编译器研究

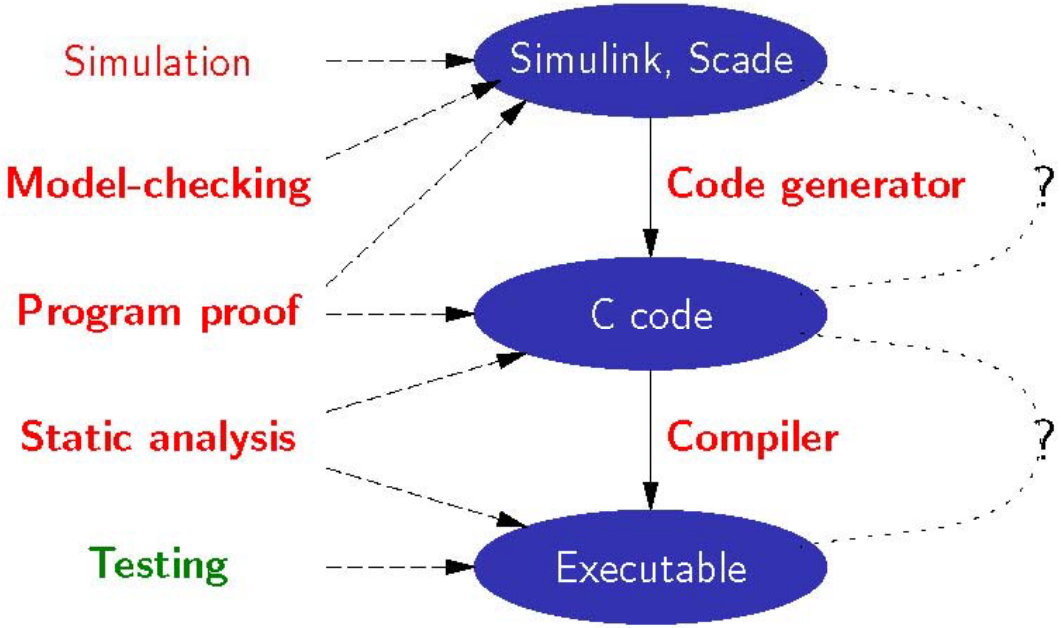
清華大學

✧ 为什么要形式化验证代码生成器/编译器

– 经过形式化验证的编译器具语义保持性

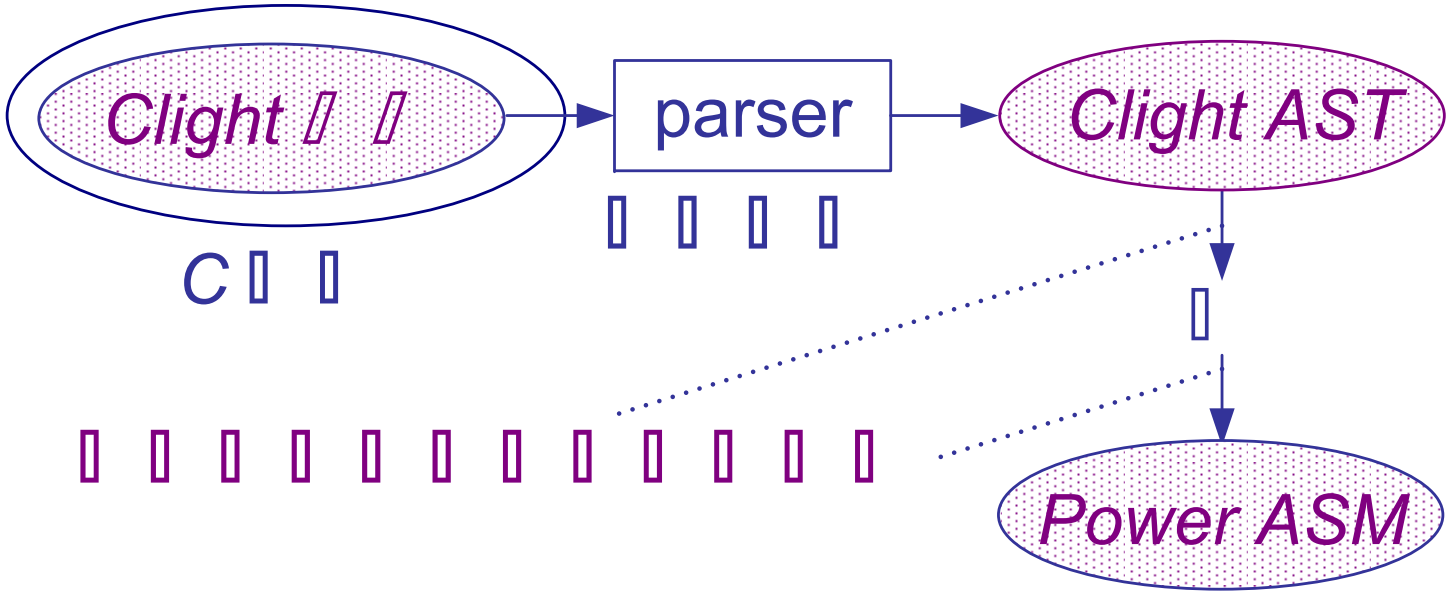
能够将源程序的行为和性质保持到所生成的目标程序

源程序级的验证工作不必要在目标程序重复



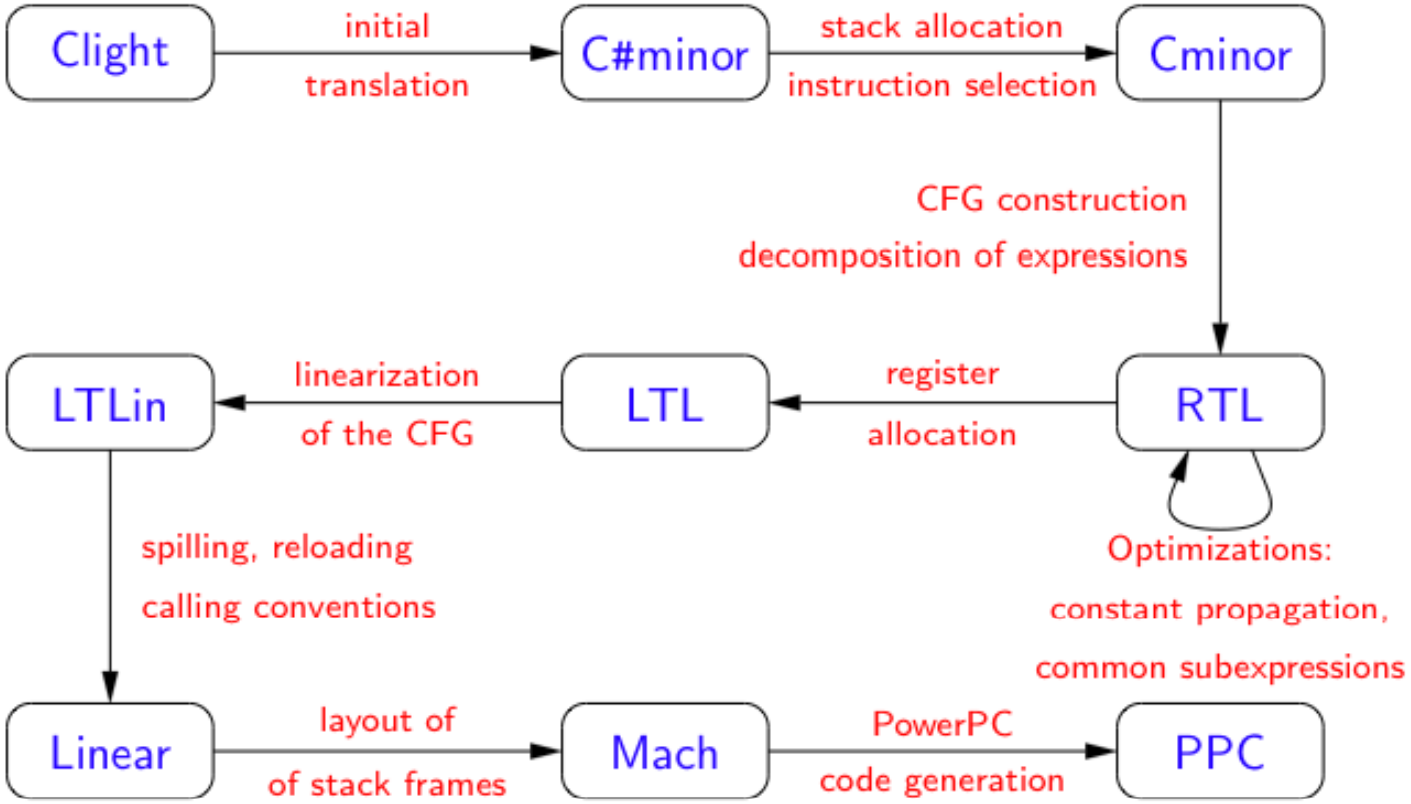
（图片源于
X. Leroy
POPL2011
特邀报告）

✧ CompCert：一个经过验证的 C 编译器



清華大學

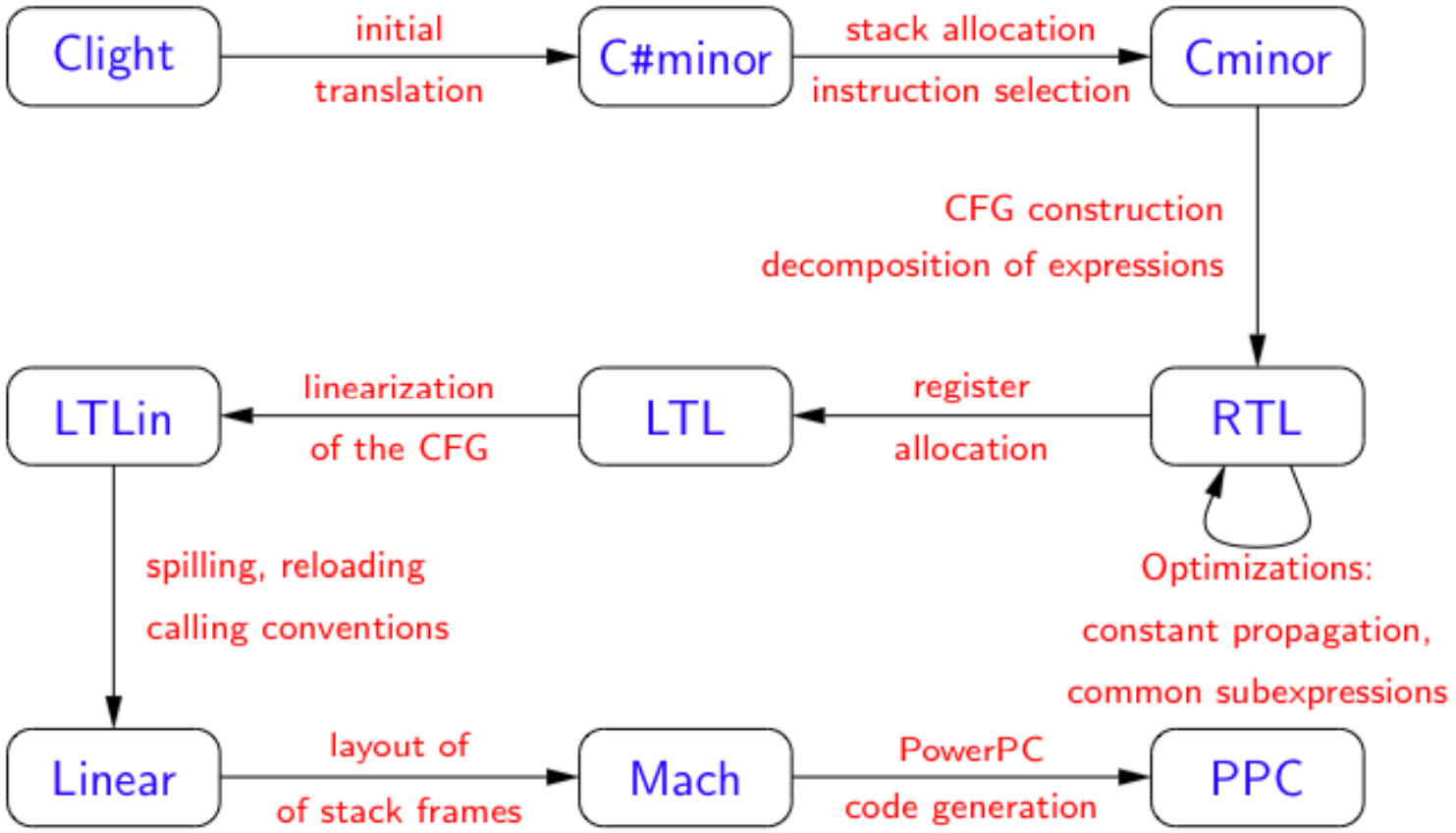
✧ CompCert: 经过验证的各编翻译过程



（图片源于 CompCert 项目）

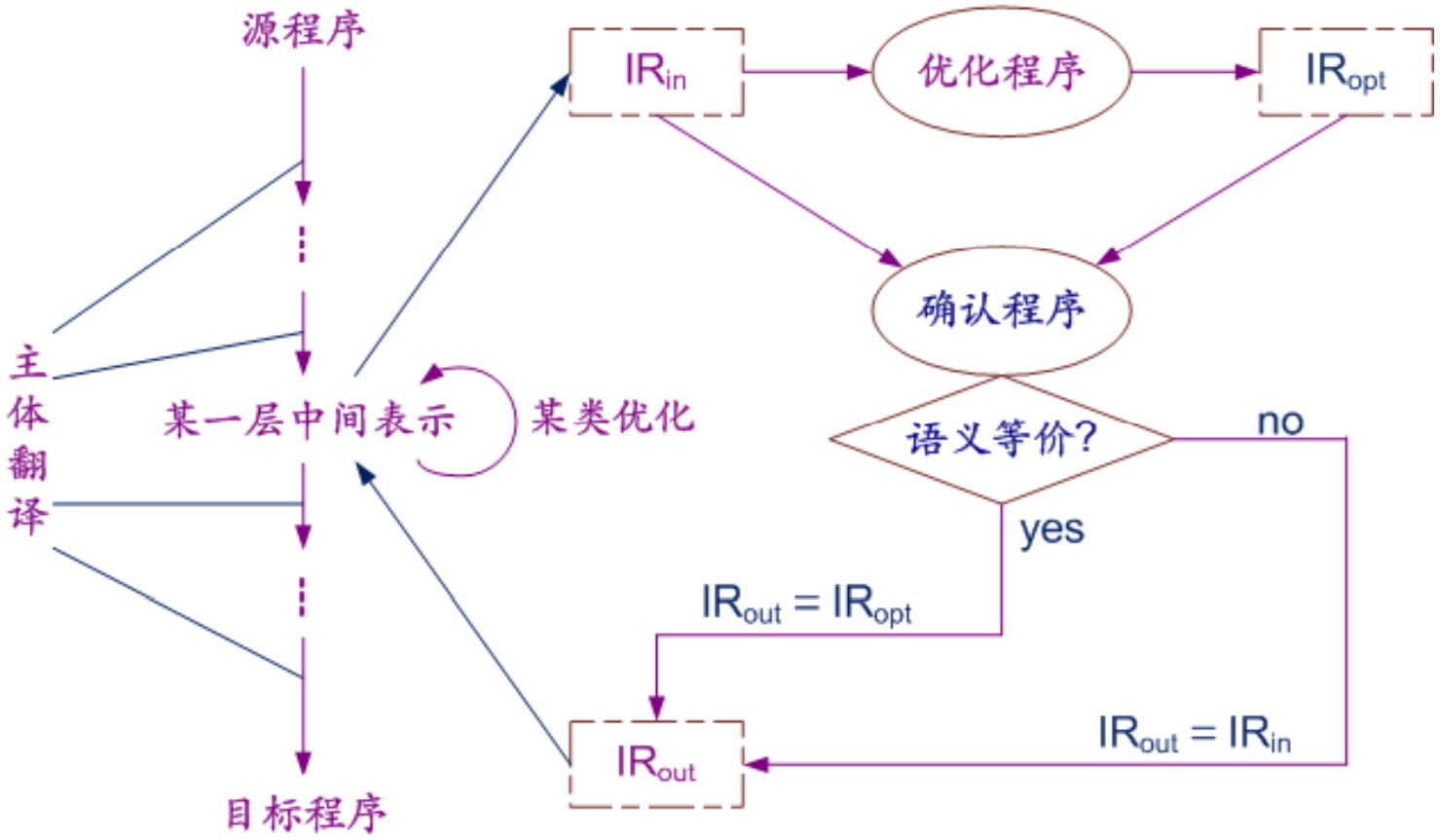
清華大學

✧ CompCert: 经过验证的各编翻译过程



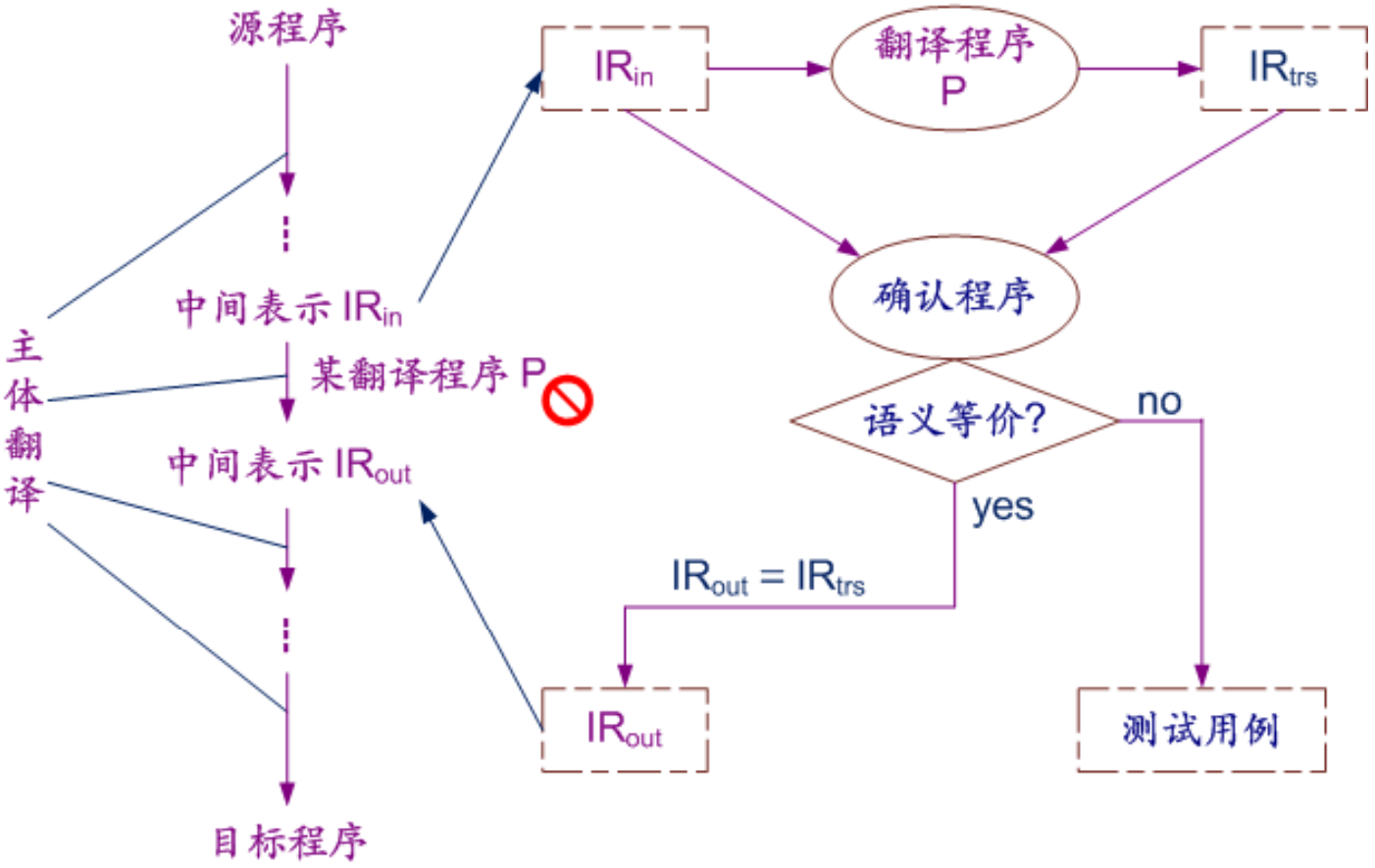
清華大學

✧ 翻译确认（*Tanslation Validation*）



清華大學

✧ 翻译确认（*Tanslation Validation*）



清華大學

✧ 同步语言

- 同步假设（*synchrony hypothesis*）

  当前周期（*cycle* 或 *instant*）的输入事件出现时，系统能够在下一周期的输入事件出现之前足够快地产生相应于当前周期的输出

- 有一定影响的同步语言

  Esterel （命令式语言）

  Lustre （陈述式语言，数据流驱动，确定性）

  Signal （陈述式语言，数据流驱动，非确定性）

清華大學

✧ 同步数据流语言 Lustre

- 流数据对象　*Streams of Values*
- 数据驱动　并发执行　逻辑时间同步
- 时态运算　*Pre*　*Fby*　*When*　*Arrow*（→）

```
node COUNTER (ck: bool) returns (x: int);
let
    x = 0 -> if ck
             then pre(x) + 1
             else pre(x)
tel
```

cycles	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	…
ck	*F*	*T*	*F*	*T*	*F*	*F*	*T*	*T*	…
X	*0*	*1*	*1*	*2*	*2*	*2*	*3*	*4*	…

清華大學

✧ 同步数据流语言 Lustre

– 时态运算　*Pre*　*Fby*　*When*　*Arrow*（→）

cycles	*0*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	…
ck	*F*	*T*	*F*	*T*	*F*	*F*	*T*	*T*	…
x	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	…
Pre (x)	*nil*	v_1	v_2	v_3	v_4	v_5	v_6	v_7	…
x when ck		v_2		v_4			v_7	v_8	…
fby (x,3,5)	5	5	5	v_1	v_2	v_3	v_4	v_5	…
$5 \rightarrow x$	5	v_2	v_3	v_4	v_5	v_6	v_7	v_8	…

同步数据流语言可信编译器研究

清華大學

✧ 同步数据流语言可信编译器

- 翻译确认（*Tanslation Validation*）
  - Signal 到 C （基于 model checking）

    A. Pnueli, M. Siegel, and O. Shtrichman （较早）

    Van Chan Ngo, etc. （近期）
- 对翻译过程本身验证
  - Lustre 到 C

    Marc Pouzet, Cédric Auger, etc.

    清华大学计算机系软件研究所系统软件与软件工程研究室

Lustre* 到 C 的可信编译器 L2C

清華大學

✧ Lustre*语言

- 支持Lustre V6 的几乎全部语言特征

  同步数据流语言 Lustre: *P.Caspi, etc., POPL'87*

  Lustre V6 : *Verimag*，法国

- Scade 扩展的多数语言特征

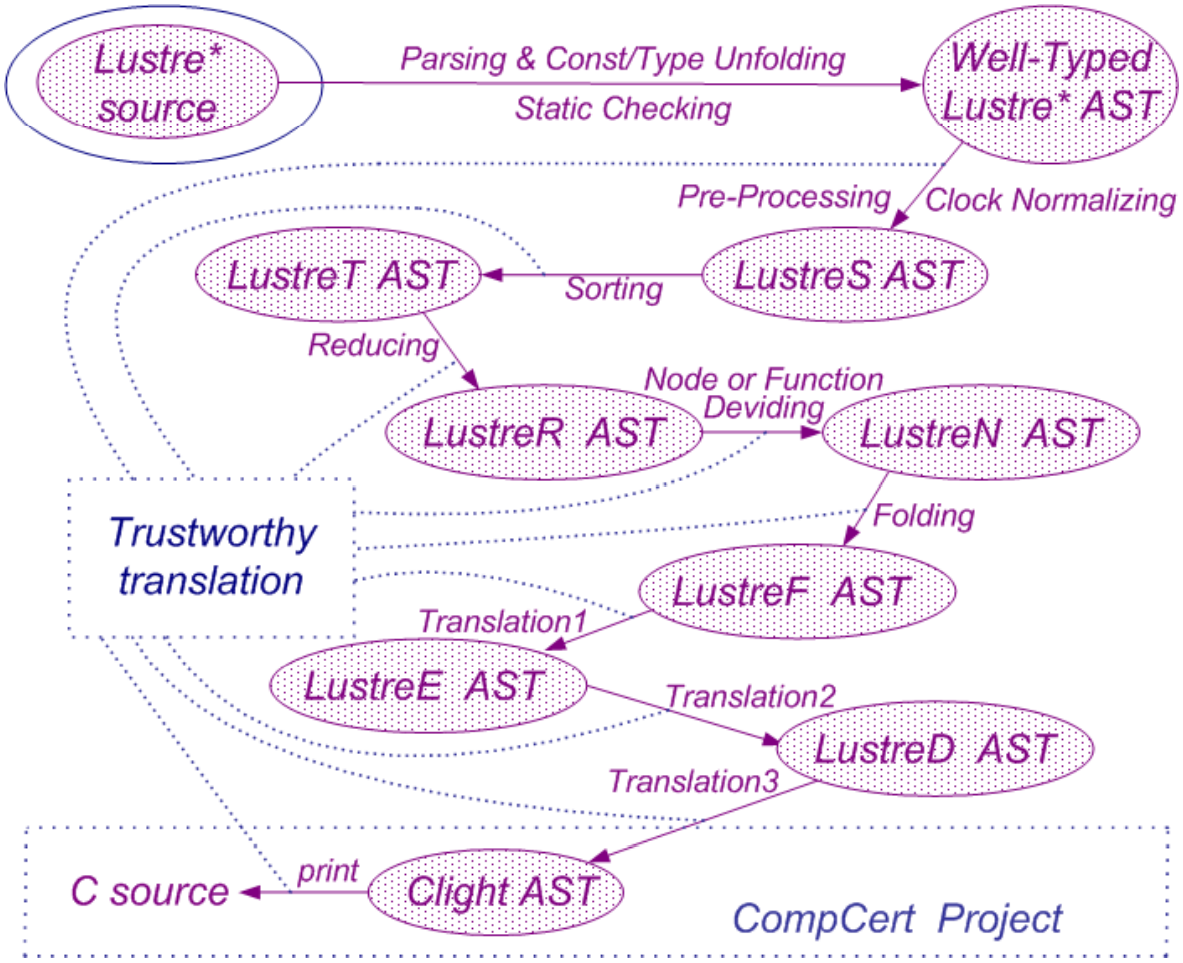
  Scade 工具：符合民航电子系统的国际标准 DO-178B（成功应用于 Airbus A340 和 A380 的设计）

  Scade 语言：Lustre 的扩展（增加了多个高阶运算，多个数组和结构体相关的运算，以个别及其他运算）

Lustre* 到 C 的可信编译器 L2C

清華大學

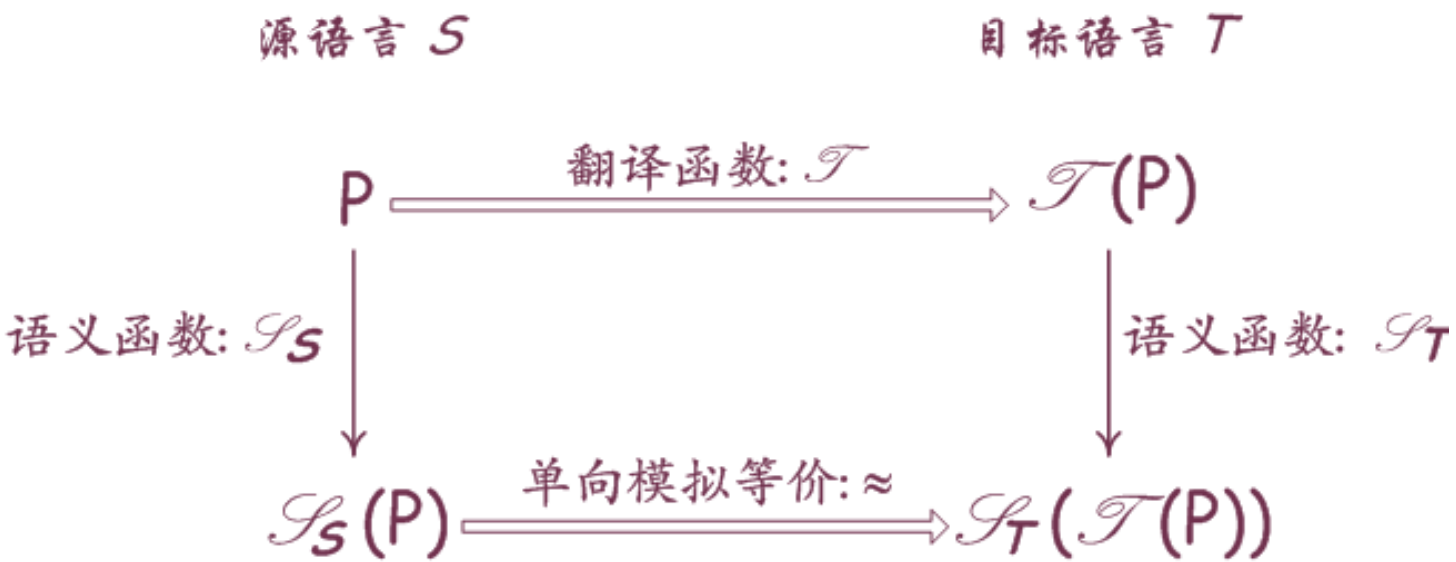
✧ 设计框架



Lustre* 到 C 的可信编译器 L2C

清華大學

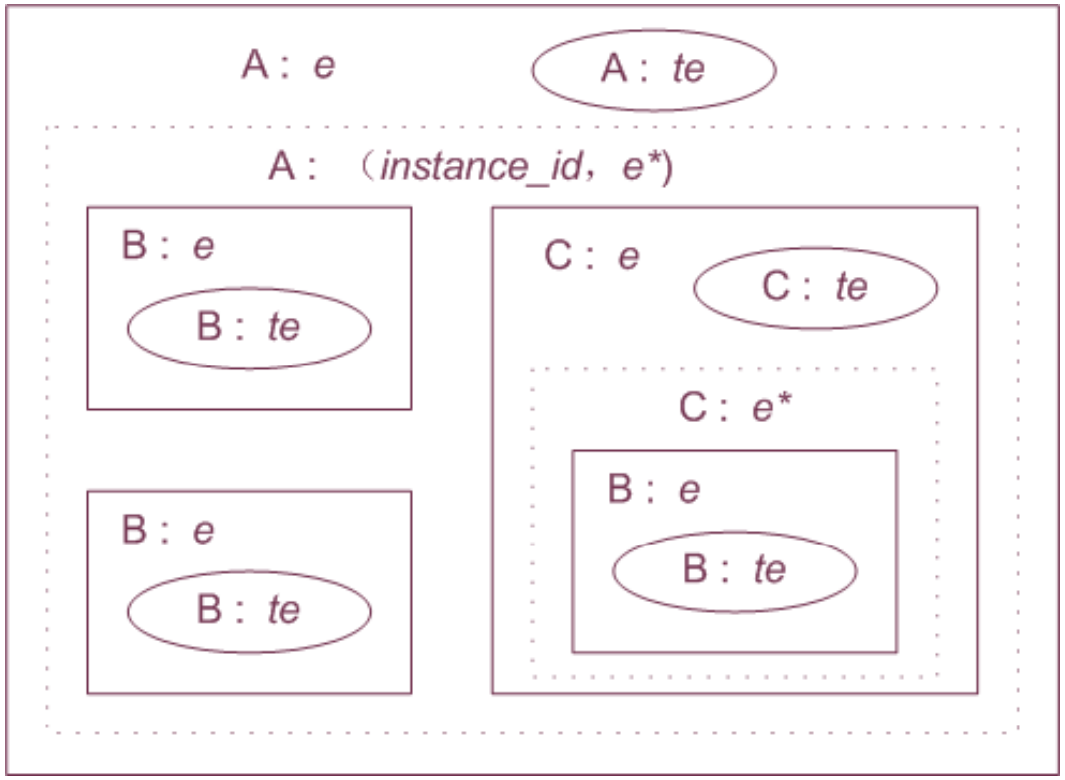
✧ 语义保持性（*Semantic Preserving*）



$$P.\ (\text{sound}(P) \Rightarrow \text{sound}(\mathscr{T}(P)) \wedge \mathscr{S}_S(P) \approx \mathscr{S}_T(\mathscr{T}(P)))$$

Lustre* 到 C 的可信编译器 L2C

✧ 语义定义中的环境（*Environment*）



$ge : (\mathcal{I}, \rho)$
$\mathcal{I} : (id \rightarrow fd)$
$\rho : (id, \tau) \rightarrow v$
$le : (id, \tau) \rightarrow v$
$te : le*$
$e : (te, id \rightarrow e*)$

欢迎宝贵建议！

谢谢！