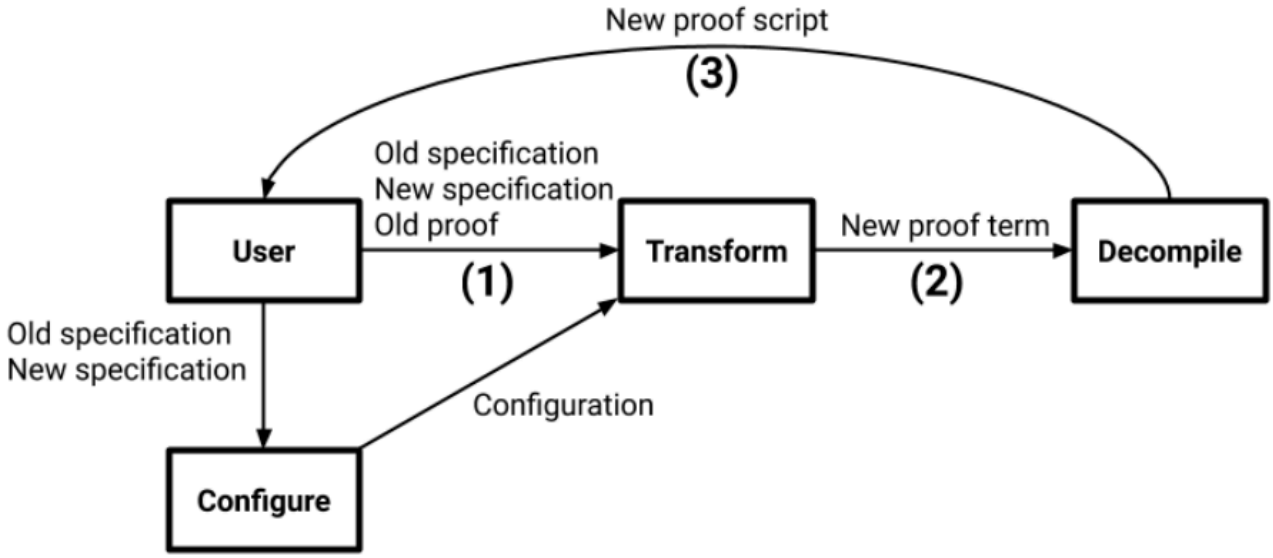
Proof Repair across Type Equivalences

2021 PLDI - talk

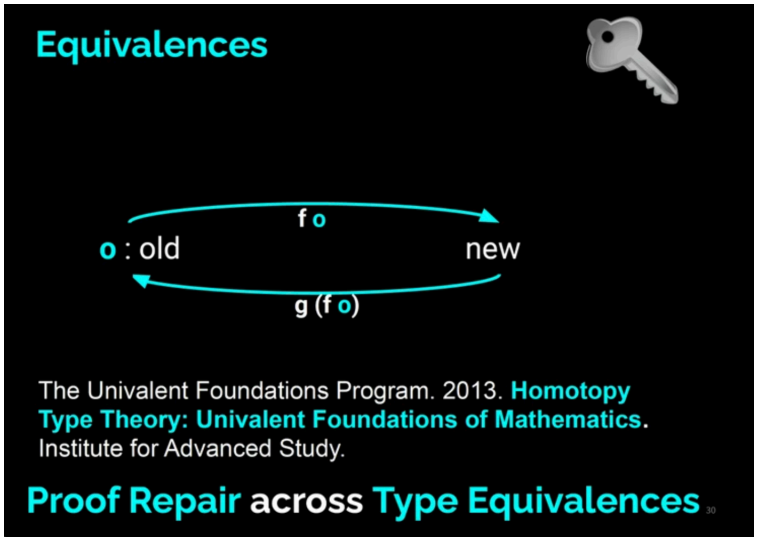
摘要：

```
在 Coq 中自动修复 “由于变换data type而损坏的证明”。

将一个 可配置的证明term的转换 和一个 从证明term到tactic scripts的反编译器 相结合。

证明term的转换：支持type等价性，删除对旧版本type的引用，不依赖于 Coq 假设之外的公理。

Pumpkin Pi（它是 Pumpkin Patch Coq 插件套件的扩展，用于验证修复）
```



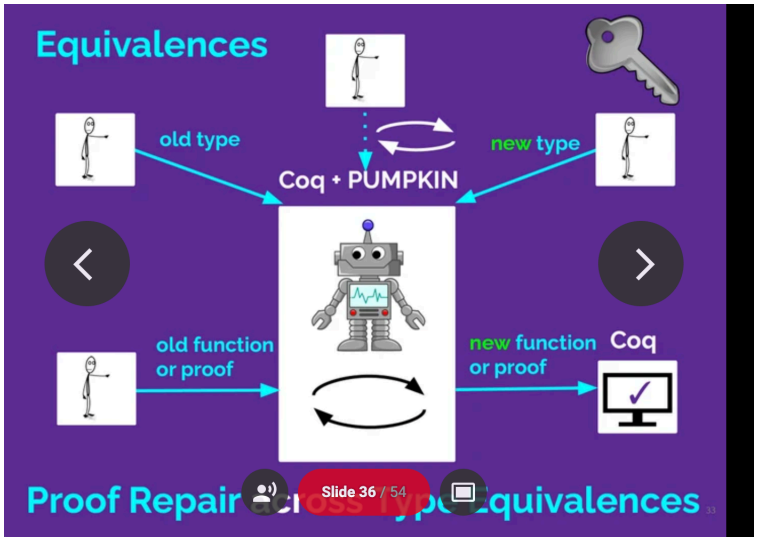
- 实现：通过在coq中加个插件实现（pumpkin pi）http://github/com/uwplse/pumpkin-pi
- 支持type equivalence的任何更改，

  用homotopy type theory描述就是，一组函数在两种type或一个type的两个版本之间的循环

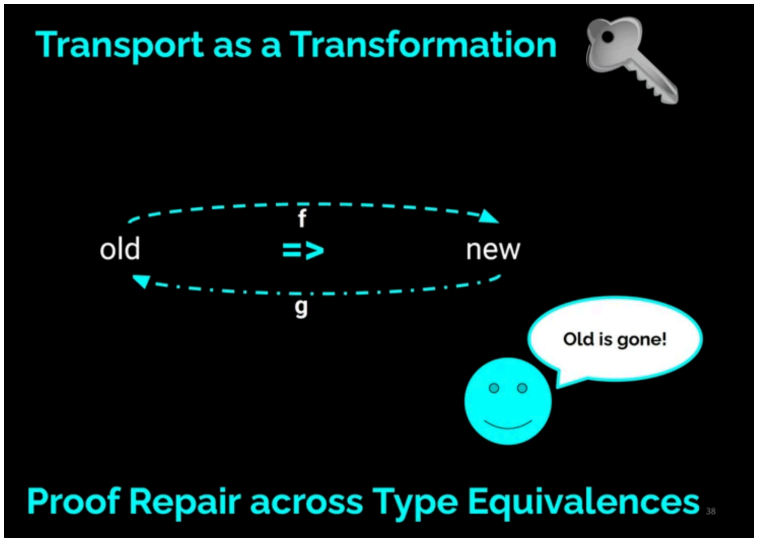


  - 有search步骤，给新旧版本的改变过的data type，会寻找并且自动证明这种描述改变的等价性。

- 当没有search时，用户可以自己pass这个equivalence。
- 最后会用新的type修复函数和证明

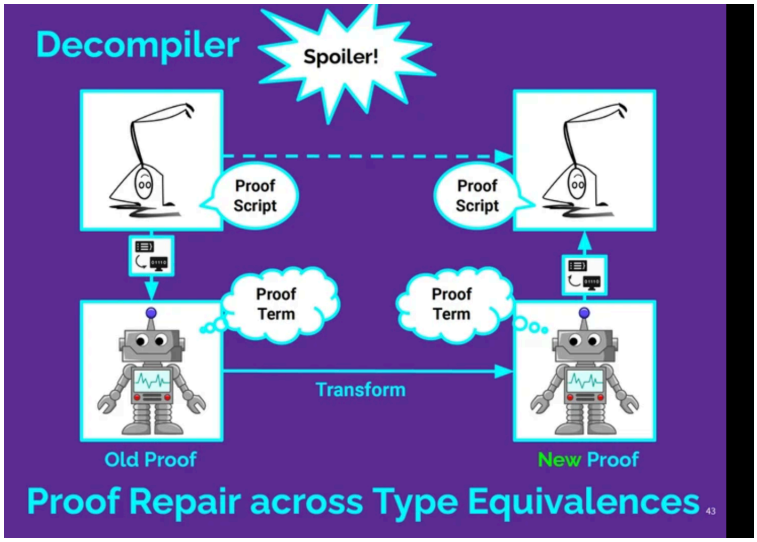


- transport：以homo理论就是通过这些equivalence来rewriting
  - 实际做的是把transform当做一个proof term transformation来实现
  - 可以在每次change之后删除旧的版本

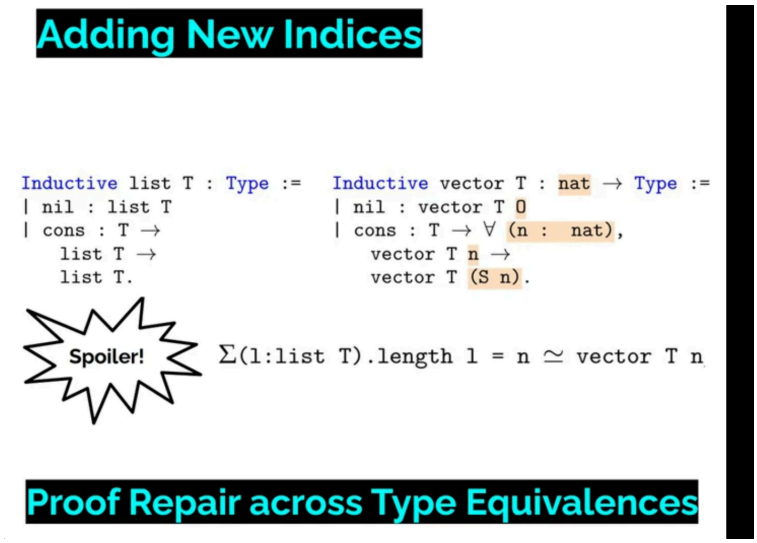


  - 把definition的等价性转换为proposition的等价性

- decompiler

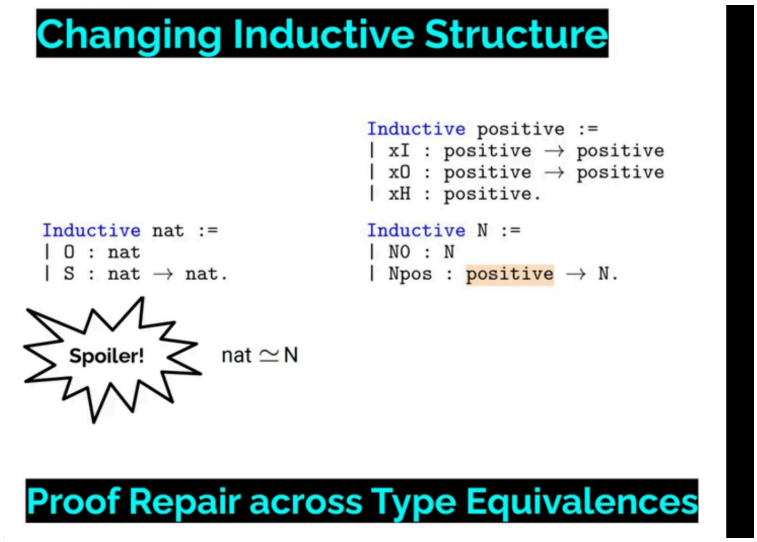


- add new indices to data type



vector 是 list 的长度索引(orange).

- change the inductive structure of data type



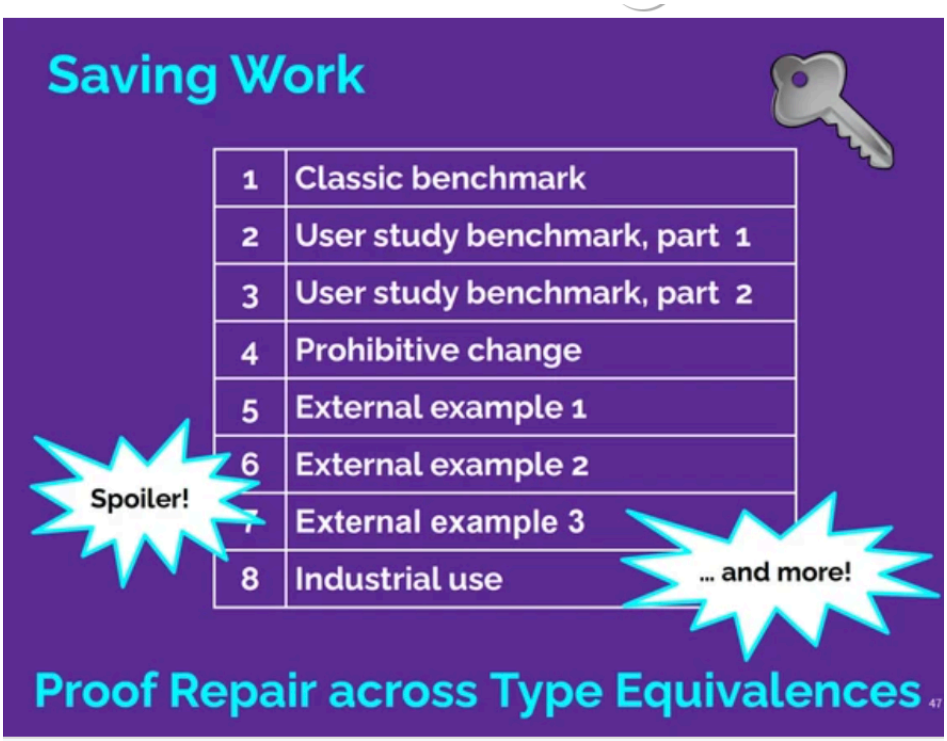
一元自然数 nat

二进制自然数 N（0:N0，Npos：positive的（1:xH、左移加1 (xI) 、左移加0 (xO)））

nat和N是等价的，但是不同的inductive structure

definitional equalities over nat -> propositional equalities over N.

- 实验：在coq标准库，25秒修复



verifying a modified program can be easier than verifying the original the first time around, in practical use cases.