Fix Generation

Yingfei Xiong, 2010
Maintain consistency between source and view automatically.
What if we need users to decide?

Source

get

put1

put2

View

?
An Example

```java
public class Test {
    public static void main(String[] args) {
        int books, publications;
        books = publications;
    }
}
```
Fixes

```java
public class Test {
    public static void main(String[] args) {
        int books, publications;
        boks = publications;
    }
}
```
Framework for Fix Generation

- \( D \), the set of data
- \( E \), the set of errors
- \( \text{check} : D \rightarrow 2^E \), the check function
- \( \text{generate} : E \rightarrow 2^{D \rightarrow D} \), the fix generating function
Law for Fix Generation

• Let

\[ e \in \text{check}(d) \]

• We have

\[ \forall f \in \text{generate}(e), \]
\[ \text{check}(f(d)) \subseteq \text{check}(d) \land \]
\[ e \notin \text{check}(f(d)) \]
Fixes are widely used

- Eclipse: Fixes for compilation errors
- [Egyed:ASE08] Fixes for UML models
- [Dallmeier:ASE09] Fixes for object behavior anomalies
- [Chen:LISA10] Fixes for firewall policy faults
- ...
Fixes are widely used

• Eclipse: Fixes for compilation errors
• [Eisenbach:ACM08] Fixes for UML model
• [Dallmeier:ASE09] Fixes for object behavior anomalies
• [Chen:LISA10] Fixes for firewall policy faults
• …

But little general support is provided
Open Issues

• How to describe check and generate?
  – using general programming languages
  – using domain-specific languages (like Boomerang / Beanbag / TGGs)

• How to ensure the law?
  – check the law
  – construct only well-defined check and generate (like Focal)

• Is it possible to construct generate automatically?
  – from consistency rules (like bidirectionalization)
  – from user operations
Bidirectional Transformation is Fix Generation with One Error

• \( D = S \times V \)
• \( E = \{ e \} \)
• \( \text{check } (s, v) | \text{get}(s) == v = \{ \} \)
  \( | \text{otherwise } = \{ e \} \)
• \( \text{generate } e (s, v) = \{ f\text{get}, f\text{put} \} \)
  – where
    • \( f\text{get} = (s, \text{get}(s)) \)
    • \( f\text{put} = (\text{put}(s,v), v) \)
Conclusion

BX → Fix Generation