



# DarcSharp

---

Darcs C# Core Library



# Agenda

- Introdução
- Implementação
  - Patch Atômico
  - Efeito
  - Inverso
  - Comutação
  - Aplicabilidade
- Verificação
- Teste
- Qualidade do Código



# Introdução

- **Objetivos:**
  - **Implementar conceitos fundamentais**
  - **Usar modelo Alloy**
  - **Verificar & Testar**



# Patch Atómico

- Utilizade?

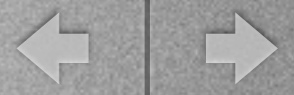
- AddFile
- AddDir
- RmFile
- RmDir
- Move
- Hunk

Alloy

```
sig Hunk extends FilePatch {  
  line : Int,  
  old : seq Line,  
  new : seq Line  
}
```

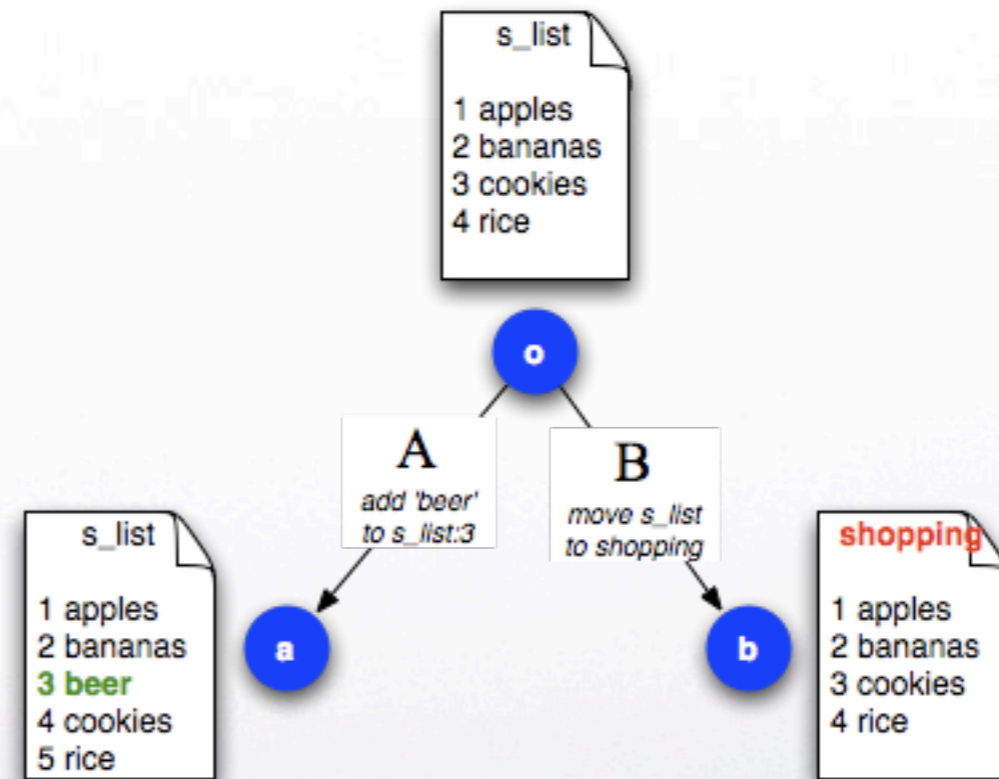
C#

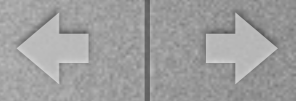
```
public class HunkPatch : PathPatch {  
  private readonly int _line;  
  private readonly List<String> _old;  
  private readonly List<String> _new;  
}
```



# Efeito

- Efeito de um Patch





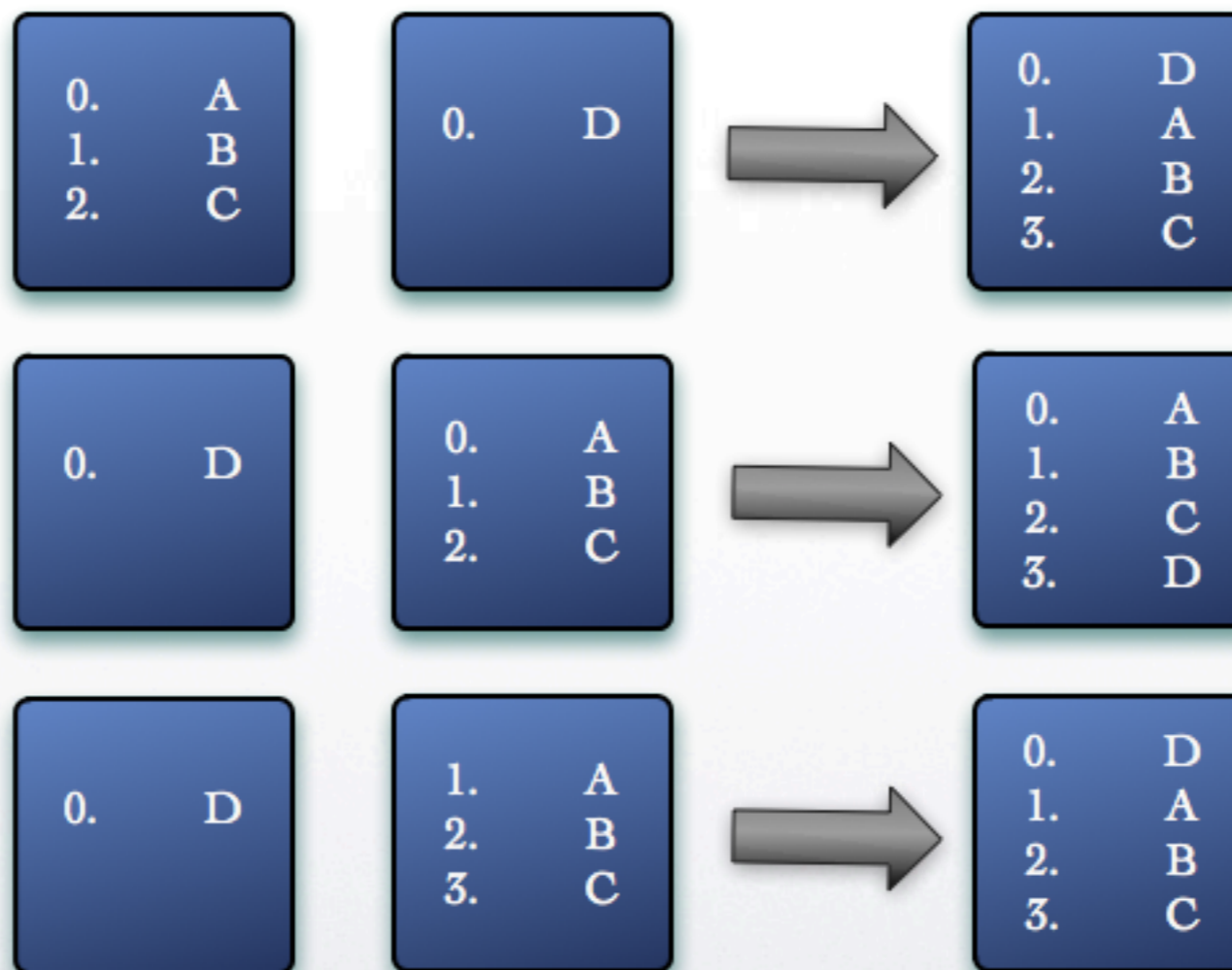
# Inverso

```
pred InvertHunk[h, h_inv : Hunk] {  
  h in Hunk and h_inv in Hunk h_inv.path = h.path  
  h_inv.line = h.line h_inv.old = h.new  
  h_inv.new = h.old  
}
```

```
public override Patch Invert() {  
  Contract.Ensures(Contract.Result<Patch>  
().Invert().Equals(this));  
  return new HunkPatch(base.Path, this.Line,  
    this.NewText, this.OldText);  
}
```

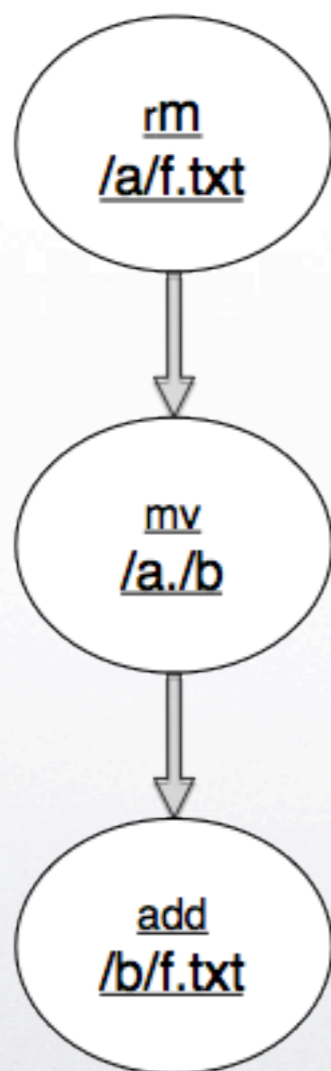


# Comutação





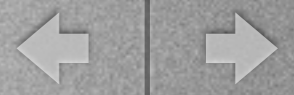
# Aplicabilidade



darcs can infer dependencies...







# Verificação

```
Contract.Ensures(Contract.ForAll(0, wrapper.GetFilesContents(this.Path).Count(), e => e >= 0 && e < this.Line &&
Contract.OldValue<String>(wrapper.GetFilesContents(this.Path)[e]).Equals(wrapper.GetFilesContents(this.Path)[e])) ||
(e >= this.Line && e < this.Line + this.NewText.Count && wrapper.GetFilesContents(this.Path)[e].Equals(this.NewText[e -
this.Line])) || (e >= this.Line + this.NewText.Count && Contract.OldValue<String>(wrapper.GetFilesContents(this.Path)[e -
this.NewText.Count + this.OldText.Count]).Equals(wrapper.GetFilesContents(this.Path)[e])));
```

```
pred ApplyHunk[t : Tree, h : Hunk, t' : Tree] {
  h in Hunk
  h.path in t.Files
  let text = t.readFile[h.path],
      old_next = h.line.add[#h.old], new_next = h.line.add[#h.new],
      old_end = old_next.prev, new_end = new_next.prev
  {
    old_end < #text and h.old = text.subseq[h.line,old_end] // old content is right
    pos[h.ldelta] and pos[#text] => text.lastIdx.add[h.ldelta] in seq/Int // respect file size limit

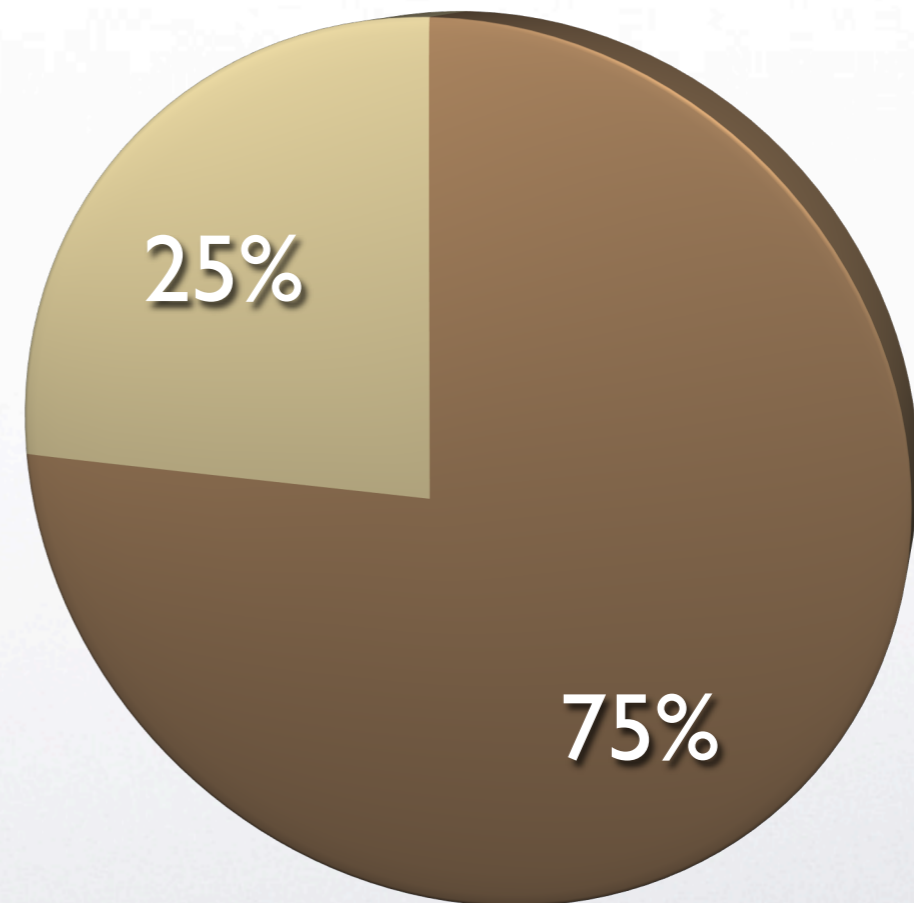
    let text' = t'.readFile[h.path] {
      WriteFile[t,h.path,text',t'] // nothing but the content of the file pointed by h.path is changed
      -- CHANGE
      #text' = (#text).add[h.ldelta]
      text'.subseq[h.line,new_end] = h.new
      -- KEEP
      text'.subseq[0,h.line.prev] = text.subseq[0,h.line.prev] // same prefix
      text'.subseq[new_next,text'.lastIdx] = text.subseq[old_next,text.lastIdx] // same rest
    }
  }
}
```



# Teste

● Tested    ● Untested

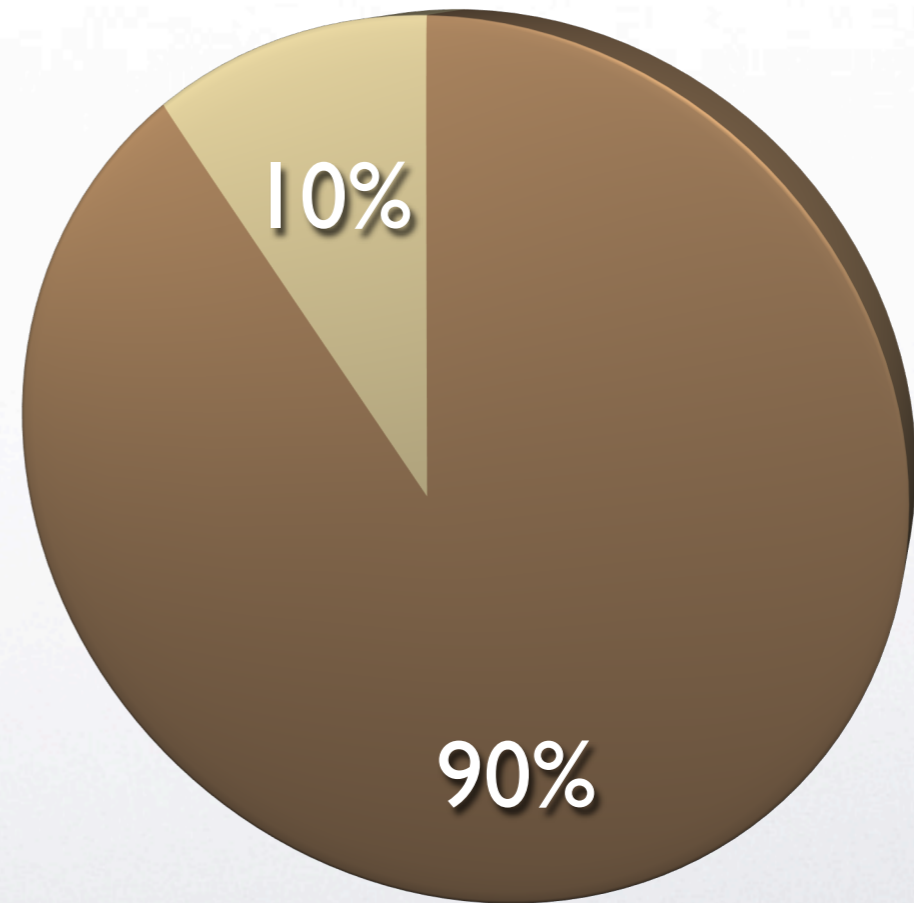
- Automated Test





# Teste

● Tested    ● Untested



- Manual Test



# Qualidade do Código

- NDepend
- CC = 3.17 (Low Risk)



# Qualidade do Código

Stat	# Occurences	Avg	StdDev	Max
Properties on interfaces	interfaces	0	0	-1 properties on
Methods on interfaces	interfaces	0	0	-1 methods on
Arguments on methods on interfaces	methods	0	0	-1 arguments on
Public properties on classes	25 Classes	0.04	0.2	1 public properties on <code>DarcSharpv2.Patches.Atomic.Generic.PathPatch</code>
Public methods on classes	25 classes	2.44	1.44	6 public methods on <code>DarcSharpv2.Patches.Atomic.Generic.Patch</code>
Arguments on public methods on classes	61 methods	0.74	0.72	4 arguments on <code>DarcSharpv2.Patches.Atomic.HunkPatch..ctor(String,Int32,List&lt;String&gt;,List&lt;String&gt;)</code>
IL instructions in non-abstract methods	82 methods	28.33	30.98	158 IL instructions in <code>DarcSharpv2.Patches.Atomic.Generic.Patch.DirectoryIsEmpty(List&lt;Patch&gt;,String,List&lt;String&gt;)</code>
Cyclomatic complexity on non abstract Methods	82 Methods	3.17	2.84	CC = 16 for <code>DarcSharpv2.Patches.Atomic.Generic.Patch.ParentExists(IEnumerable&lt;Patch&gt;,String)</code>



# Conclusão

- Objectivos cumpridos:
  - Implementação do Core
  - Testes exaustivos usando Pex
  - Transposição do modelo Alloy para C#



# Obrigado

- **Questões?**