Symbolic Execution of Debian Packages

Nicolas Jeannerod
nicolas.jeannerod@irif.fr

joint work with Benedikt Becker, Claude Marché
Yann Régis-Gianas, Mihaela Sighireanu, Ralf Treinen

IRIF, Université de Paris

September 9, 2019

13th Alpine Verification Meeting
Introduction

- CoLiS project: Correctness of Linux Scripts
CoLiS project: Correctness of Linux Scripts

Goal: applying formal methods to the quality assessment of Debian Packages.
Introduction

- CoLiS project: Correctness of Linux Scripts

- Goal: applying formal methods to the quality assessment of Debian Packages.

- Debian: operating system.

- Packages: way to provide (install, update, remove) software.

Goal (reformulated): making sure that installing/updating/removing software does not:
- make other softwares unusable,
- make the whole computer unusable,
- remove your personal files,
- etc.
Introduction

> CoLiS project: Correctness of Linux Scripts

> Goal: applying formal methods to the quality assessment of Debian Packages.

> Debian: operating system.

> Packages: way to provide (install, update, remove) software.

> Goal (reformulated): making sure that installing/updating/removing software does not:
  > make other softwares unusable,
  > make the whole computer unusable,
  > remove your personal files,
  > etc.
Installing a Software on Debian

1. Download the package.
Installing a Software on Debian

1. Download the package.

2. Execute a pre-installation script.
1. Download the package.

2. Execute a pre-installation script.

3. Unpack static archive.
1. Download the package.
2. Execute a pre-installation script.
3. Unpack static archive.
4. Execute a post-installation script.
1. Download the package.

2. Execute a pre-installation script.
   > This is a POSIX shell script ran as administrator.

3. Unpack static archive.

4. Execute a post-installation script.
   > This is a POSIX shell script ran as administrator.
1. Download the package.

2. Execute a pre-installation script.
   > This is a POSIX shell script ran as administrator.

3. Unpack static archive.

4. Execute a post-installation script.
   > This is a POSIX shell script ran as administrator.

---

POSIX shell:

> scripting language
Installing a Software on Debian

1. Download the package.

2. Execute a pre-installation script.
   > This is a POSIX shell script ran as administrator.

3. Unpack static archive.

4. Execute a post-installation script.
   > This is a POSIX shell script ran as administrator.

POSIX shell:

> scripting language

> legacy (born in 1971)
Installing a Software on Debian

1. Download the package.

2. Execute a pre-installation script.
   > This is a POSIX shell script ran as administrator.

3. Unpack static archive.

4. Execute a post-installation script.
   > This is a POSIX shell script ran as administrator.

---

**POSIX shell:**

> scripting language

> legacy (born in 1971)

**Administrator:**

> can do anything on the system
Installing a Software on Debian

1. Download the package.

2. Execute a pre-installation script.
   > This is a POSIX shell script ran as administrator.

3. Unpack static archive.

4. Execute a post-installation script.
   > This is a POSIX shell script ran as administrator.

POSIX shell:
> scripting language
> legacy (born in 1971)

Administrator:
> can do anything on the system

Complicated and dangerous
Installing a Software on Debian

1. Download the package.

2. Execute a pre-installation script.
   > This is a POSIX shell script ran as administrator.

3. Unpack static archive.

4. Execute a post-installation script.
   > This is a POSIX shell script ran as administrator.

---

POSIX shell:
> scripting language
> legacy (born in 1971)

Administrator:
> can do anything on the system

Complicated and dangerous. Formal methods?
Our Tools: An Overview

Debian Package

CoLiS

Report
Our Tools: An Overview

Debian Package

Shell script

CoLiS

Specification of the script

Symbolic Engine

Report
Our Tools: An Overview

CoLiS

Debian Package

Shell script

Morbig, Morsmall and ColisFromShell

Colis inter. language

Specification of the script

Symbolic Engine

Report
Our Tools: An Overview

Debian Package

Morbig, Morsmall and ColisFromShell

Symbolic Engine

Specifications of commands

Shell script

Colis inter. language

Specification of the script

Report

CoLiS
Our Tools: An Overview

Debian Package

CoLiS

Shell script

Specification of the script

Morbig, Morsmall and ColisFromShell

Colis inter. language

Symbolic Engine

Specifications of commands

SAT?  

SAT solver for specifications

Report
Our Tools: An Overview

Debian Package

Shell script

Specification of the script

Morbig, Morsmall and ColisFromShell

Colis inter. language

Symbolic Engine

Specifications of commands

SAT solver for specifications

Report

Régis-Gianas, J & Treinen
SLE 2018

J, Marché & Treinen
VSTTE 2017
Specifications,
Feature Trees & Constraints
Feature Trees

> Unranked unordered trees;
Feature Trees

- Unranked unordered trees;
- Good models for the UNIX filesystem;
Feature Trees

Unranked unordered trees;

Good models for the UNIX filesystem;

Shell scripts can be seen as programs that modify such trees;
Unranked unordered trees;

Good models for the UNIX filesystem;

Shell scripts can be seen as programs that modify such trees;

Constraints will express relations between such trees.
Atom (Informal) Semantics

From $x$'s tree, through $f$, we go to $y$'s tree.

In $x$'s tree, there is no $f$.

The root of $x$'s tree has decoration $A$.

$x$'s tree can also use features in $F$.

$x$ and $y$'s trees are similar except in $F$. 
Atom (Informal) Semantics

\[ x[f]y \] From \( x \)'s tree, through \( f \), we go to \( y \)'s tree

\[ x[f]^\uparrow \] In \( x \)'s tree, there is no \( f \)

\[ Ax \] The root of \( x \)'s tree has decoration \( A \)

[ Aït-Kaci, Podelski & Smolka 1992 ]
Constraints On Feature Trees

Atom (Informal) Semantics

\[ x[f]y \quad \text{From } x\text{’s tree, through } f, \text{ we go to } y\text{’s tree} \]

\[ x[f]^\uparrow \quad \text{In } x\text{’s tree, there is no } f \]

\[ Ax \quad \text{The root of } x\text{’s tree has decoration } A \]

\[ x[F] \quad x\text{’s tree can also use features in } F \]
Atom (Informal) Semantics

\( x[f]y \) From \( x \)'s tree, through \( f \), we go to \( y \)'s tree

\( x[f]↑ \) In \( x \)'s tree, there is no \( f \)

\( Ax \) The root of \( x \)'s tree has decoration \( A \)

\( x[F] \) \( x \)'s tree can also use features in \( F \)

\( x \sim_F y \) \( x \) and \( y \)'s trees are similar except in \( F \)
Example Specification: mkdir q/f

\[ \exists x, x', y'. \]
\[ \text{resolve}(r, \text{cwd}, q, x) \wedge \text{dir}(x) \wedge x[f]↑ \]
\[ \wedge \text{similar}(r, r', \text{cwd}, q, x, x') \wedge x \sim \{f\} x' \]
\[ \wedge \text{dir}(x') \wedge x'[f]y' \wedge \text{dir}(y') \wedge y'[\emptyset] \]

Success

\[ \exists y \cdot \text{resolve}(r, \text{cwd}, q/f, y) \wedge r \models r' \]

Error

\[ \exists x \cdot \text{resolve}(r, \text{cwd}, q, x) \wedge \neg \text{dir}(x) \wedge r \models r' \]
Example Specification: mkdir q/f

\[ \exists x, x', y'. \]

\[ \text{resolve}(r, cwd, q, x) \land \text{dir}(x) \land x[f] \uparrow \]

\[ \land \text{similar}(r, r', cwd, q, x, x') \land x \sim \{f\} x' \]

\[ \land \text{dir}(x') \land x'[f]y' \land \text{dir}(y') \land y'[\emptyset] \]

---

\[ \exists y \cdot \text{resolve}(r, cwd, q/f, y) \land y \sim \{f\} y' \]

---

\[ \text{noresolve}(r, cwd, q) \land r = r' \]

---

\[ \exists x \cdot \text{resolve}(r, cwd, q, x) \land \neg \text{dir}(x) \]
Example Specification: mkdir q/f

\[\exists x, x', y'.\]

(resolve(r, cwd, q, x) \land \text{dir}(x) \land x[f] \uparrow
\land \text{similar}(r, r', cwd, q, x, x') \land x \sim \{f\} x'
\land \text{dir}(x') \land x'[f] y' \land \text{dir}(y') \land y'[\emptyset]
\]

Success

\[\exists y \cdot \text{resolve}(r, cwd, q/f, y) \land y[f] \uparrow\]

noresolve(r, cwd, q) \land r \doteq\]

\[\exists x \cdot \text{resolve}(r, cwd, q, x) \land \neg \text{dir}(x)\]
Example Specification: mkdir q/f

\[ \exists x, x', y'. \]

\[ \text{resolve}(r, \text{cwd}, q, x) \land \text{dir}(x) \land x[f] \uparrow \land \text{similar}(r, r', \text{cwd}, q, x, x') \land x \sim\{f\} x' \land \text{dir}(x') \land x'[f]y' \land \text{dir}(y') \land y'[\emptyset] \]

\[ \exists y \cdot \text{resolve}(r, \text{cwd}, q/f, y) \land y \sim\{f\} x' \]

\[ \text{noresolve}(r, \text{cwd}, q) \land r \hat{=} \]

\[ \exists x \cdot \text{resolve}(r, \text{cwd}, q, x) \land \neg \text{dir}(x) \]
Example Specification: \texttt{mkdir q/f}

\exists x, x', y'. 

\texttt{resolve}(r, \texttt{cwd}, q, x) \land \texttt{dir}(x) \land x[f] \uparrow \\
\land \texttt{similar}(r, r', \texttt{cwd}, q, x, x') \land x \sim \{f\} x' \\
\land \texttt{dir}(x') \land x'[f]y' \land \texttt{dir}(y') \land y'[\emptyset]

\texttt{Success}

\exists y \cdot \texttt{resolve}(r, \texttt{cwd}, q/f, y) \land \texttt{dir}(y) \land y[f] \downarrow

\texttt{noresolve}(r, \texttt{cwd}, q) \land r \models

\exists x \cdot \texttt{resolve}(r, \texttt{cwd}, q, x) \land \neg \texttt{dir}(x) \\
\models
Example Specification: mkdir q/f

$$\exists x, x', y'.$$

$$\text{resolve}(r, \text{cwd}, q, x) \wedge \text{dir}(x) \wedge x[f] \uparrow$$

$$\wedge \text{similar}(r, r', \text{cwd}, q, x, x') \wedge x \sim \{f\} x'$$

$$\wedge \text{dir}(x') \wedge x'[f]y' \wedge \text{dir}(y') \wedge y'[\emptyset]$$

\[\text{Success}\]

$$\exists y \cdot \text{resolve}(r, \text{cwd}, q/f, y) \wedge r \overset{(\text{dir})}{\sim}$$

\[\text{noresolve}(r, \text{cwd}, q) \wedge r \overset{(\text{dir})}{\not\sim}\]

$$\exists x \cdot \text{resolve}(r, \text{cwd}, q, x) \wedge \neg \text{dir}(x)$$
Example Specification: mkdir q/f

\[\exists x, x', y'.\]

\[\text{resolve}(r, \text{cwd}, q, x) \land \text{dir}(x) \land x[f] \uparrow\]
\[\land \text{similar}(r, r', \text{cwd}, q, x, x') \land x \sim \{f\} x'\]
\[\land \text{dir}(x') \land x'[f]y' \land \text{dir}(y') \land y'[\emptyset]\]

\[\exists y \cdot \text{resolve}(r, \text{cwd}, q/f, y) \land \text{dir}(y)\]

\[\text{noresolve}(r, \text{cwd}, q) \land r \dashv\]

\[\exists x \cdot \text{resolve}(r, \text{cwd}, q, x) \land \neg \text{dir}(x)\]
Example Specification: mkdir q/f

∃x, x', y'.

\[\text{resolve}(r, \text{cwd}, q, x) \land \text{dir}(x) \land x[f] \uparrow \]
\[\land \text{similar}(r, r', \text{cwd}, q, x, x') \land x \sim \{f\} x' \]
\[\land \text{dir}(x') \land x'[f]y' \land \text{dir}(y') \land y'[\emptyset] \]

\[\exists y \cdot \text{resolve}(r, \text{cwd}, q/f, y) \land r' \]
\[\text{noresolve}(r, \text{cwd}, q) \land r \]
\[\exists x \cdot \text{resolve}(r, \text{cwd}, q, x) \land \neg \text{dir}(x) \]

\[r \cdots \sim \{q\} \cdots r' \]
\[q \]
\[\exists x \cdots \sim \{f\} \cdots \exists x' \]
\[f \]
\[\perp \]
Example Specification: mkdir q/f

\[ \exists x, x', y'. \]

\[ \text{resolve}(r, \text{cwd}, q, x) \land \text{dir}(x) \land x[f] \uparrow \land \text{similar}(r, r', \text{cwd}, q, x, x') \land x \sim \{f\} x' \land \text{dir}(x') \land x'[f]y' \land \text{dir}(y') \land y'[\emptyset] \]

Success

\[ \exists y \cdot \text{resolve}(r, \text{cwd}, q/f, y) \land \text{noresolve}(r, \text{cwd}, q) \land r \vdash \]

\[ \exists x \cdot \text{resolve}(r, \text{cwd}, q, x) \land \neg \text{dir}(x) \]

\[ r \ldots \sim \{q\} \ldots r' \]
\[ q \[
\exists x \ldots \sim \{f\} \ldots \exists x' \]
\[ f \[
\exists y' \]
Example Specification: mkdir q/f

\[ \exists x, x', y'. \]

\[
\text{resolve}(r, \text{cwd}, q, x) \land \text{dir}(x) \land x[f] \uparrow \\
\land \text{similar}(r, r', \text{cwd}, q, x, x') \land x \sim \{f\} x' \\
\land \text{dir}(x') \land x'[f]y' \land \text{dir}(y') \land y'[\emptyset]
\]

Success

\[ \exists y \cdot \text{resolve}(r, \text{cwd}, q/f, y) \land y[f] \downarrow \]

noresolve(r, cwd, q) \land r \Downarrow

\[ \exists x \cdot \text{resolve}(r, \text{cwd}, q, x) \land \neg \text{dir}(x) \land r \Downarrow \]

\[ \exists x \cdot \text{resolve}(r, \text{cwd}, q, x) \land \neg \text{dir}(x) \land r \Downarrow \]

\[ \exists y' \cdot \text{empty dir} \]

\[ r \overset{\sim}{\rightarrow} \{q\} \overset{\sim}{\rightarrow} r' \]

\[ q \]

\[ \exists x \overset{\sim}{\rightarrow} \{f\} \overset{\sim}{\rightarrow} \exists x' \]

\[ f \]

\[ \exists y' \overset{(\emptyset \text{ dir})}{\rightarrow} \]

\[ \exists x \overset{(\text{dir})}{\rightarrow} \]

\[ \exists y' \overset{(\text{empty dir})}{\rightarrow} \]
Symbolic Execution
Symbolic Execution

```bash
if [ -e foo ]; then
    rm foo
fi
```
Symbolic Execution

```sh
if [ -e foo ]; then
  rm foo
fi
```

In progress

```
In progress

r
```
Symbolic Execution

```bash
if [ -e foo ]; then
    rm foo
fi
```

---

**Case 1**
**Success**

```
r = r'
  foo
  /
  └─ 
```

---

**In progress**

```
r
  foo
  /
  └─ x
```
Symbolic Execution

```bash
if [ -e foo ]; then
  rm foo
fi
```

---

Case 1
Success

```
r = r'
foo
⊥
```

Case 2
Success

```
r ≈_{foo} r'
foo
X (¬dir)
⊥
```

Case 3
Error

```
r = r'
foo
X (dir)
```
Chaining Specifications

mkdir /usr/lib ; mkdir /usr/lib/foo
Chaining Specifications

```plaintext
mkdir /usr/lib ; mkdir /usr/lib/foo
```

```
usr
\downarrow
lib
```

```
usr
\downarrow
\{usr\}
```

```
lib
\downarrow
\{lib\}
```

```
x_1 \sim \{lib\} \cdots x_1'
```

```
usr
\downarrow
\{usr\}
```

```
r_1 \cdots \sim \{usr\} \cdots r_1'
```

```
usr
\downarrow
\{usr\}
```

```
y_1'[\emptyset]
```

[J & Treinen, IJCAR 2018]
Chaining Specifications

mkdir /usr/lib ; mkdir /usr/lib/foo

$r_1 \cdots \sim \{\text{usr}\} \cdots r_1'$

usr

\hline

usr

$\sim\{\text{lib}\}$

\hline

lib

\hline

lib

\hline

$y_1'[\emptyset]$

\hline

$y_2'[\emptyset]$

\hline

foo

\hline

foo

\hline

$z_2'[\emptyset]$
Chaining Specifications

mkdir /usr/lib ; mkdir /usr/lib/foo

\[
\begin{array}{c}
\text{usr} \quad \sim \{\text{usr}\} \\
x_1 \quad \sim \{\text{lib}\} \\
\text{lib} \\
\downarrow \\
y_1'[\emptyset]
\end{array}
\quad \quad \quad
\begin{array}{c}
\text{usr} \\
x_1' \\
\text{lib} \\
\downarrow \\
y_1'[\emptyset]
\end{array}
\quad \quad \quad
\begin{array}{c}
\text{usr} \quad \sim \{\text{usr}\} \\
x_2 \quad \sim \{\text{lib}\} \\
\text{lib} \\
\downarrow \\
y_2'[\emptyset]
\end{array}
\quad \quad \quad
\begin{array}{c}
\text{usr} \quad \sim \{\text{lib}\} \\
x_2' \\
\text{lib} \\
\downarrow \\
y_2'[\emptyset]
\end{array}
\]

[J & Treinen, IJCAR 2018]
mkdir /usr/lib ; mkdir /usr/lib/foo

\[
\begin{array}{c}
\text{usr}  \\
x_1 \sim \{\text{lib}\}  \\
\downarrow  \\
\text{lib}  \\
\end{array}
\quad
\begin{array}{c}
\text{usr}  \\
x_1' \sim \emptyset  \\
\downarrow  \\
\text{y}_1'\emptyset  \\
\end{array}
\quad
\begin{array}{c}
\text{usr}  \\
x_2 \sim \{\text{lib}\}  \\
\downarrow  \\
\text{lib}  \\
\end{array}
\quad
\begin{array}{c}
\text{usr}  \\
x_2' \sim \emptyset  \\
\downarrow  \\
\text{y}_2'\emptyset  \\
\end{array}
\quad
\begin{array}{c}
\text{usr}  \\
x_{12} \sim \{\text{usr}\}  \\
\downarrow  \\
\text{usr}  \\
\end{array}
\quad
\begin{array}{c}
\text{usr}  \\
x_{12}' \sim \emptyset  \\
\downarrow  \\
\text{z}_2'\emptyset  \\
\end{array}
\end{array}
\]
Chaining Specifications

```
mkdir /usr/lib  ;  mkdir /usr/lib/foo
```

![Diagram of Chaining Specifications]
Chaining Specifications

mkdir /usr/lib ; mkdir /usr/lib/foo

\[
\begin{align*}
& \quad r_1 \cdots \sim \{\text{usr}\} \cdots r_{12} \cdots \sim \{\text{usr}\} \cdots r'_2 \\
& \quad \text{usr} \quad \cdots \sim \{\text{lib}\} \cdots \text{usr} \quad \cdots \sim \{\text{lib}\} \cdots \text{usr} \\
& \quad x_1 \quad \cdots \sim \{\text{lib}\} \cdots x_{12} \quad \cdots \sim \{\text{lib}\} \cdots x'_2 \\
& \quad \text{lib} \quad \cdots \sim \{\text{foo}\} \cdots \text{lib} \quad \cdots \sim \{\text{foo}\} \cdots \text{lib} \\
& \quad y_{12}[\emptyset] \quad \cdots \sim \{\text{foo}\} \cdots y'_2 \\
& \quad \text{usr} \quad \cdots \sim \{\text{lib}\} \cdots \text{usr} \quad \cdots \sim \{\text{lib}\} \cdots \text{usr} \\
& \quad z'_2[\emptyset]
\end{align*}
\]
Chaining Specifications

mkdir /usr/lib ; mkdir /usr/lib/foo

\[ r_1 \sim \{\text{usr}\} \quad r_{12} \sim \{\text{usr}\} \quad r_2' \quad \sim \{\text{usr}\} \\]

\[ \sim \{\text{lib}\} \quad \sim \{\text{lib}\} \quad \sim \{\text{lib}\} \quad \sim \{\text{lib}\} \quad \sim \{\text{foo}\} \quad \sim \{\text{foo}\} \quad \sim \{\text{foo}\} \quad \sim \{\text{foo}\} \\]

\[ y_{12}[\emptyset] \quad y_2' \quad z_2'[\emptyset] \\]

\[ x_1 \quad x_{12} \quad x_2' \]\n
\[ \begin{align*}
usr & \quad \sim \{\text{usr}\} & \quad \sim \{\text{usr}\} & \quad \sim \{\text{usr}\} & \quad \sim \{\text{usr}\} \\
lib & \quad \sim \{\text{lib}\} & \quad \sim \{\text{lib}\} & \quad \sim \{\text{lib}\} & \quad \sim \{\text{lib}\} \\
fo \o & \quad \sim \{\text{foo}\} & \quad \sim \{\text{foo}\} & \quad \sim \{\text{foo}\} & \quad \sim \{\text{foo}\} \\
\emptyset & \quad \sim \{\emptyset\} & \quad \sim \{\emptyset\} & \quad \sim \{\emptyset\} & \quad \sim \{\emptyset\} \\
\end{align*} \]

\[ \text{J & Treinen, IJCAR 2018} \]
mkdir /usr/lib ; mkdir /usr/lib/foo
mkdir /usr/lib ; mkdir /usr/lib/foo

[J & Treinen, IJCAR 2018]
Report > oz

Meta
Start time
2019-07-20 21:41:15
End time
2019-07-20 21:41:15
Duration
0s

Parsing
Name
oz
Version
0.16.0-2
Maintainer scripts
postinst
OK
prerm
Rejected by conversion unsupported feature: (word_component)
postrm
OK
Installation Scenario

Scenarios

Installation

- preinst install
  - OK
  - Failed

  unpack
  - OK
  - Failed

  postinst configure
    - OK
    - Failed

  unsup. utility
  - Installed
    - (0 states)
  - Failed-Config
    - (0 states)

  postrm abort-install
  - OK
  - Failed

  Not-Installed
    - (0 states)
  Half-Installed
    - (0 states)

Removal
An Other Scenario

Idempotency of `postrm purge`

- `postrm remove`
  - OK: OSEF (1 states)
  - Failed: Non-Idempotent (0 states)

- `postrm remove`
  - OK: OSEF (0 states)
  - Failed: Non-Idempotent (0 states)

- `postrm purge`
  - OK: OSEF (2 states)
  - Failed: Non-Idempotent (0 states)

- `postrm purge`
  - OK: Non-Idempotent (1 states)
  - Failed: OSEF (0 states)

OSEF
0 1
Non-Idempotent
0
An Execution Case

log

[UTL] test 'purge' = 'purge': strings are equal
[UTL] test -f /etc/oz/id_rsa-icicle-gen: path resolves to file of type 'f'
[UTL] rm /etc/oz/id_rsa-icicle-gen: remove file
[UTL] rm /etc/oz/id_rsa-icicle-gen.pub: target does not exist or is a directory
[UTL] test -f /etc/oz/id_rsa-icicle-gen.pub: target does not exist or is a directory
The postrm Script

Original Shell script

```bash
#!/bin/sh

set -e

FILE="/etc/oz/id_rsa-icicle-gen"

case "$1" in
   purge)
      if [ -f $FILE ]; then
         rm $FILE $FILE.pub
      fi
   ;;
   remove|upgrade|failed-upgrade|abort-install|abort-upgrade|disappear)
      ;;
   *)
      echo "postrm called with unknown argument \"$1\"" >&2
         exit 1
   ;;
   esac

# dh_installdeb will replace this with shell code automatically
# generated by other debhelper scripts.
```
Conclusion

> Demo report accessible from my website:
> http://nicolas.jeannerod.fr/

> CoLiS project: Correctness of Linux Script.
> Webpage: http://colis.irif.fr/
> Tools: https://github.com/colis-anr/

So far, 148 bugs found and reported to Debian;
Several talks at DebConf;
The Debian maintainers are very enthusiastic!

Future work: support more packages
Support more shell constructs,
Add more command specifications,
Improve the constraint solver;

Thank you for your attention!
Conclusion

> Demo report accessible from my website: http://nicolas.jeannerod.fr/

> CoLiS project: Correctness of Linux Script.
  > Webpage: http://colis.irif.fr/
  > Tools: https://github.com/colis-anr/

So far, 148 bugs found and reported to Debian;
Several talks at DebConf;
The Debian maintainers are very enthusiastic!

Future work: support more packages
Support more shell constructs,
Add more command specifications,
Improve the constraint solver;

Thank you for your attention!
Conclusion

> Demo report accessible from my website: http://nicolas.jeannerod.fr/

> CoLiS project: Correctness of Linux Script.
  > Webpage: http://colis.irif.fr/
  > Tools: https://github.com/colis-anr/

> So far, 148 bugs found and reported to Debian;
> Several talks at DebConf;
  The Debian maintainers are very enthusiastic!
Conclusion

> Demo report accessible from my website: http://nicolas.jeannerod.fr/

> CoLiS project: Correctness of Linux Script.
  > Webpage: http://colis.irif.fr/
  > Tools: https://github.com/colis-anr/

> So far, 148 bugs found and reported to Debian;

> Several talks at DebConf;
  The Debian maintainers are very enthusiastic!

> Future work: support more packages
  > Support more shell constructs,
  > Add more command specifications,
  > Improve the constraint solver;
Conclusion

> Demo report accessible from my website: http://nicolas.jeannerod.fr/

> CoLiS project: Correctness of Linux Script.
  > Webpage: http://colis.irif.fr/
  > Tools: https://github.com/colis-anr/

> So far, 148 bugs found and reported to Debian;

> Several talks at DebConf;
  The Debian maintainers are very enthusiastic!

> Future work: support more packages
  > Support more shell constructs,
  > Add more command specifications,
  > Improve the constraint solver;

> Thank you for your attention!