# 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

# Introduction

- 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.

# 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 personnal 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.

# Installing a Software on Debian

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

# Installing a Software on Debian

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

# 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.

# 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

# 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)

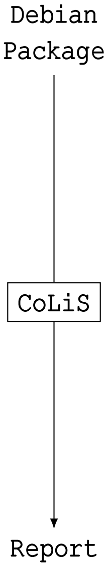
Administrator:

> can do anything on the system

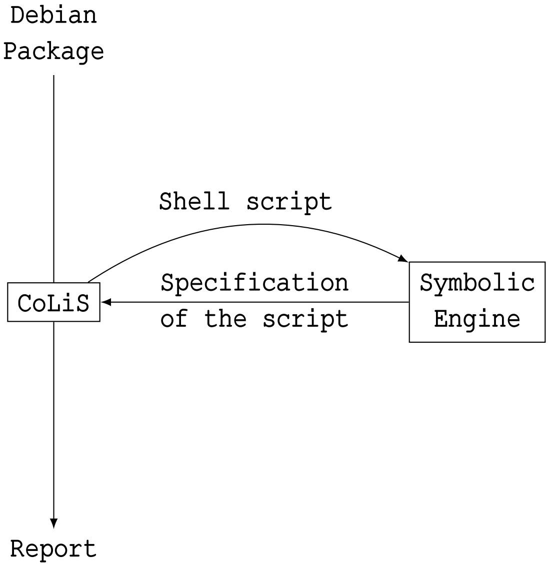
---

Complicated and dangerous. Formal methods?

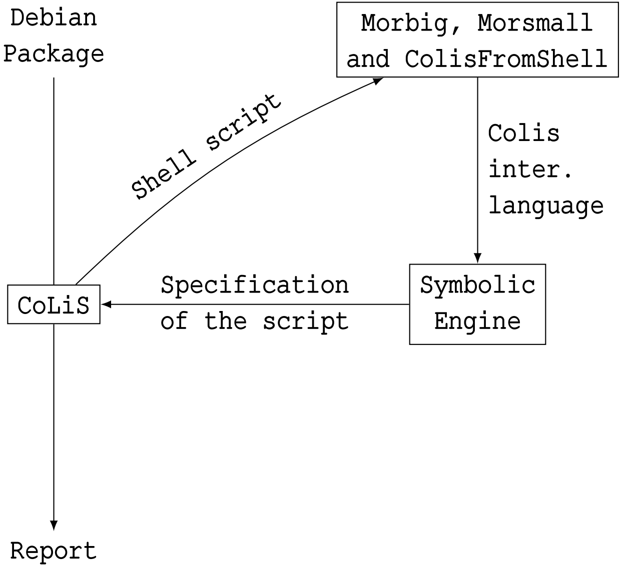
# Our Tools: An Overview



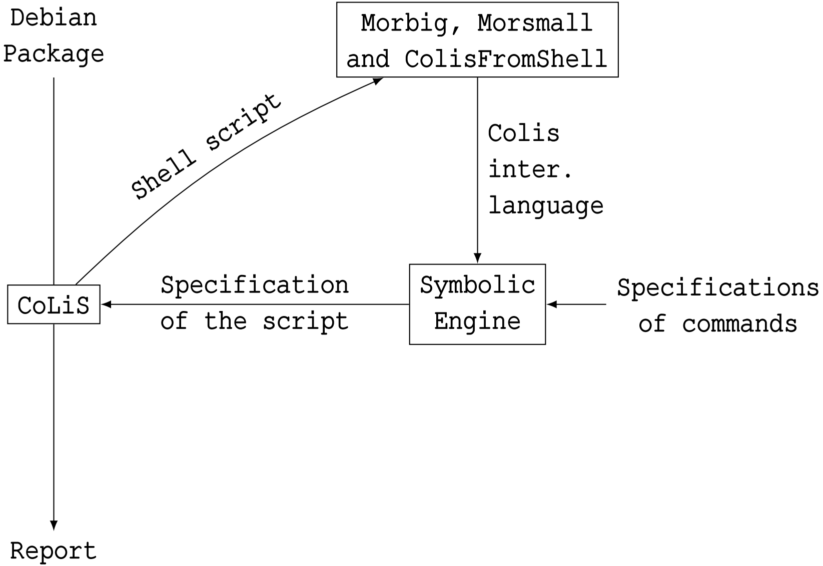
# Our Tools: An Overview



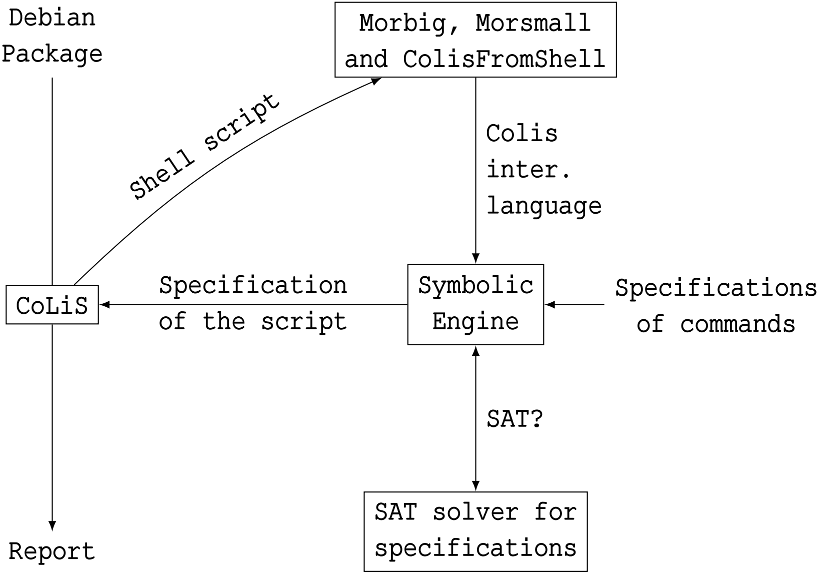
# Our Tools: An Overview



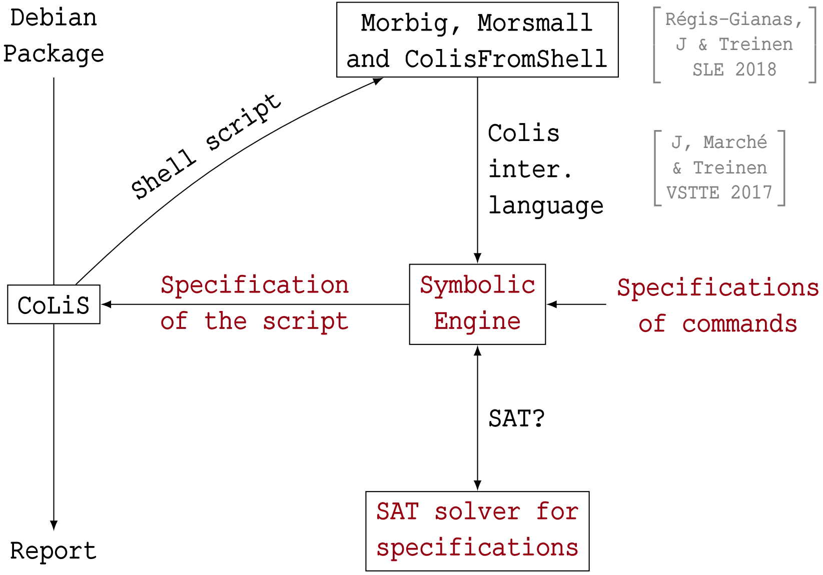
# Our Tools: An Overview



# Our Tools: An Overview

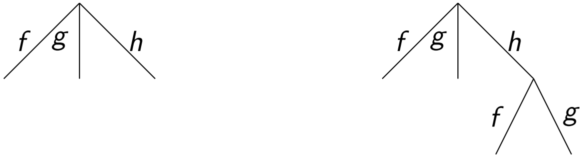

Debian Package
Morbig, Morsmall and ColisFromShell
Shell script
Colis inter. language
CoLiS
Specification of the script
Symbolic Engine
Specifications of commands
SAT?
Report
SAT solver for specifications

# Our Tools: An Overview



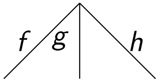
# Specifications,
# Feature Trees & Constraints

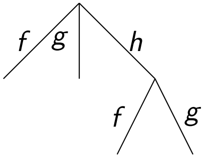
# Feature Trees



- Unranked unordered trees;

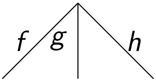
# Feature Trees

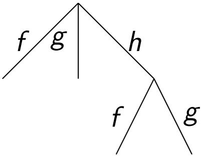




- Unranked unordered trees;
- Good models for the UNIX filesystem;

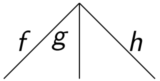
# Feature Trees

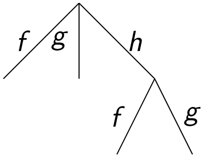




- Unranked unordered trees;
- Good models for the UNIX filesystem;
- Shell scripts can be seen as programs that modify such trees;

# Feature 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.

# Constraints On Feature Trees

| Atom | (Informal) Semantics |
|---|---|

# Constraints On Feature Trees

| 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 | |
|---|---|---|
| $x[f]y$ | From $x$'s tree, through $f$, we go to $y$'s tree | |
| $x[f]\uparrow$ | In $x$'s tree, there is no $f$ | [Aït-Kaci Podelski & Smolka 1992] |
| $Ax$ | The root of $x$'s tree has decoration $A$ | |
| $x[F]$ | $x$'s tree can also use features in $F$ | [Smolka & Treinen 1994] |

# Constraints On Feature Trees

| 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 $f$ | [Aït-Kaci Podelski & Smolka 1992] |
| $Ax$ | The root of $x$'s tree has decoration $A$ | |
| $x[F]$ | $x$'s tree can also use features in $F$ | [Smolka & Treinen 1994] |
| $x \sim_F y$ | $x$ and $y$'s trees are similar except in $F$ | |

# Example Specification: mkdir q/f

$$\begin{aligned}&\exists x, x', y' \cdot \\ &\mathtt{resolve}(r, cwd, q, x) \land \mathtt{dir}(x) \land x[f]\uparrow \\ &\land \mathtt{similar}(r, r', cwd, q, x, x') \land x \sim_{\{f\}} x' \\ &\land \mathtt{dir}(x') \land x'[f]y' \land \mathtt{dir}(y') \land y'[\varnothing]\end{aligned}$$

Success

---

$$\exists y \cdot \mathtt{resolve}(r, cwd, q/f, y) \land r \doteq r'$$

---

$$\mathtt{noresolve}(r, cwd, q) \land r \doteq r'$$

Error

---

$$\exists x \cdot \mathtt{resolve}(r, cwd, q, x) \land \neg \mathtt{dir}(x) \land r \doteq r'$$

# Example Specification: mkdir q/f

$\exists x, x', y' \cdot$

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

$\wedge\, \texttt{similar}(r, r', cwd, q, x, x') \wedge x \sim_{\{f\}} x'$

$\wedge\, \texttt{dir}(x') \wedge x'[f]y' \wedge \texttt{dir}(y') \wedge y'[\varnothing]$

Success

$\exists y \cdot \texttt{resolve}(r, cwd, q/f, y) \wedge$

$\texttt{noresolve}(r, cwd, q) \wedge r \doteq$

$\exists x \cdot \texttt{resolve}(r, cwd, q, x) \wedge \neg \texttt{dir}($

# Example Specification: mkdir q/f

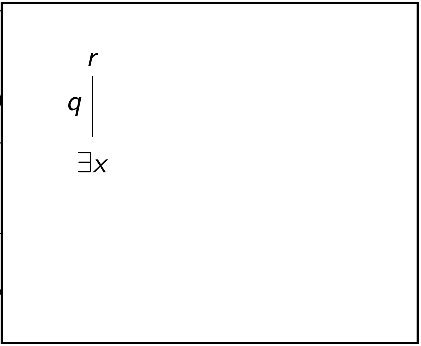
$$\begin{aligned}
&\exists x, x', y' \cdot \\
&\texttt{resolve}(r, cwd, q, x) \wedge \texttt{dir}(x) \wedge x[f]\uparrow \\
&\wedge \texttt{similar}(r, r', cwd, q, x, x') \wedge x \sim_{\{f\}} x' \\
&\wedge \texttt{dir}(x') \wedge x'[f]y' \wedge \texttt{dir}(y') \wedge y'[\varnothing]
\end{aligned}$$

Success

$\exists y \cdot \texttt{resolve}(r, cwd, q/f, y) \wedge r$

$\texttt{noresolve}(r, cwd, q) \wedge r \doteq$

$\exists x \cdot \texttt{resolve}(r, cwd, q, x) \wedge \neg \texttt{dir}(x$



# Example Specification: mkdir q/f

$\exists x, x', y' \cdot$

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

$\wedge\, \texttt{similar}(r, r', cwd, q, x, x') \wedge x \sim_{\{f\}} x'$

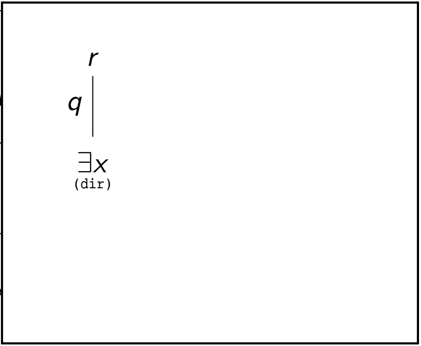
$\wedge\, \texttt{dir}(x') \wedge x'[f]y' \wedge \texttt{dir}(y') \wedge y'[\varnothing]$

Success

$\exists y \cdot \texttt{resolve}(r, cwd, q/f, y) \wedge$

$\texttt{noresolve}(r, cwd, q) \wedge r \doteq$

$\exists x {\cdot} \texttt{resolve}(r, cwd, q, x) {\wedge} \neg \texttt{dir}($



# Example Specification: mkdir q/f

$\exists x, x', y' \cdot$

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

$\wedge\, \mathtt{similar}(r, r', cwd, q, x, x') \wedge x \sim_{\{f\}} x'$

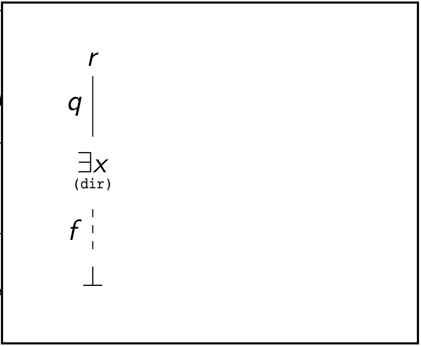
$\wedge\, \mathtt{dir}(x') \wedge x'[f]y' \wedge \mathtt{dir}(y') \wedge y'[\varnothing]$

Success

$\exists y \cdot \mathtt{resolve}(r, cwd, q/f, y) \wedge$

$\mathtt{noresolve}(r, cwd, q) \wedge r \doteq$

$\exists x \cdot \mathtt{resolve}(r, cwd, q, x) \wedge \neg \mathtt{dir}($



# Example Specification: mkdir q/f

$$\exists x, x', y' \cdot$$

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

$$\wedge\ \texttt{similar}(r, r', cwd, q, x, x') \wedge x \sim_{\{f\}} x'$$

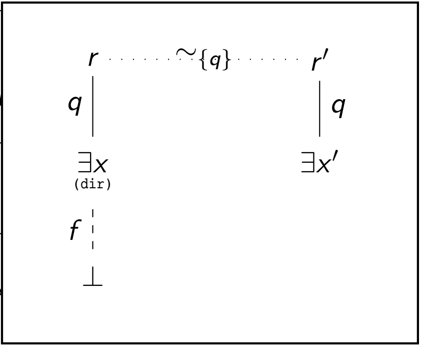
$$\wedge\ \texttt{dir}(x') \wedge x'[f]y' \wedge \texttt{dir}(y') \wedge y'[\varnothing]$$

Success

$\exists y \cdot \texttt{resolve}(r, cwd, q/f, y) \wedge$

$\texttt{noresolve}(r, cwd, q) \wedge r \doteq$

$\exists x \cdot \texttt{resolve}(r, cwd, q, x) \wedge \neg\texttt{dir}($



# Example Specification: mkdir q/f

$\exists x, x', y' \cdot$

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

$\wedge\, \texttt{similar}(r, r', cwd, q, x, x') \wedge x \sim_{\{f\}} x'$

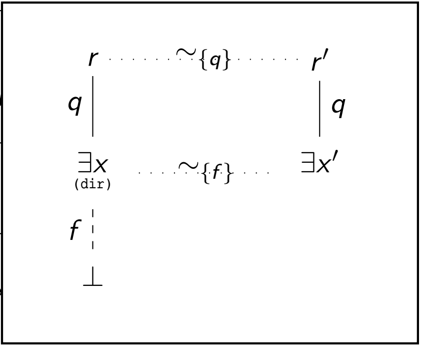
$\wedge\, \texttt{dir}(x') \wedge x'[f]y' \wedge \texttt{dir}(y') \wedge y'[\varnothing]$

Success

$\exists y \cdot \texttt{resolve}(r, cwd, q/f, y) \wedge$

$\texttt{noresolve}(r, cwd, q) \wedge r \doteq$

$\exists x \cdot \texttt{resolve}(r, cwd, q, x) \wedge \neg \texttt{dir}($



# Example Specification: mkdir q/f

$$\exists x, x', y' \cdot$$

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

$$\wedge\, \texttt{similar}(r, r', cwd, q, x, x') \wedge x \sim_{\{f\}} x'$$

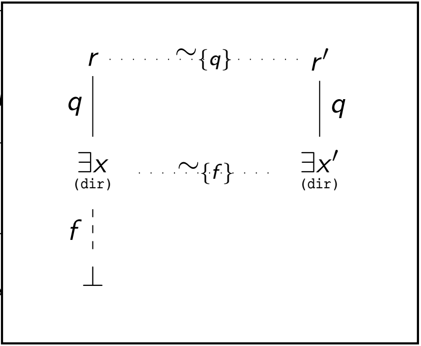
$$\wedge\, \texttt{dir}(x') \wedge x'[f]y' \wedge \texttt{dir}(y') \wedge y'[\varnothing]$$

Success

$\exists y \cdot \texttt{resolve}(r, cwd, q/f, y) \wedge$

$\texttt{noresolve}(r, cwd, q) \wedge r \doteq$

$\exists x \cdot \texttt{resolve}(r, cwd, q, x) \wedge \neg \texttt{dir}($



# Example Specification: mkdir q/f

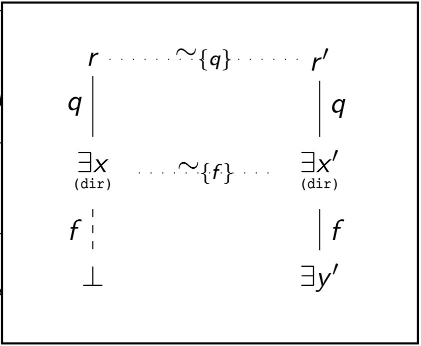
$$\begin{aligned}
&\exists x, x', y' \cdot \\
&\texttt{resolve}(r, cwd, q, x) \wedge \texttt{dir}(x) \wedge x[f]\uparrow \\
&\wedge \texttt{similar}(r, r', cwd, q, x, x') \wedge x \sim_{\{f\}} x' \\
&\wedge \texttt{dir}(x') \wedge x'[f]y' \wedge \texttt{dir}(y') \wedge y'[\varnothing]
\end{aligned}$$

Success

$\exists y \cdot \texttt{resolve}(r, cwd, q/f, y) \wedge r$

$\texttt{noresolve}(r, cwd, q) \wedge r \doteq$

$\exists x \cdot \texttt{resolve}(r, cwd, q, x) \wedge \neg \texttt{dir}(x$



# Example Specification: mkdir q/f

$$\exists x, x', y' \cdot$$

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

$$\land\, \texttt{similar}(r, r', cwd, q, x, x') \land x \sim_{\{f\}} x'$$

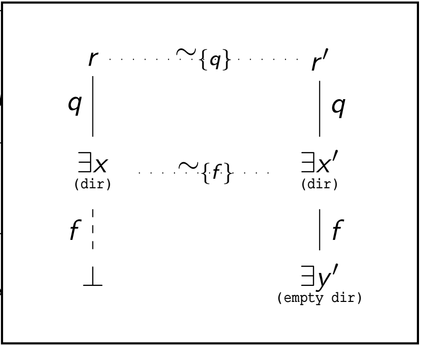
$$\land\, \texttt{dir}(x') \land x'[f]y' \land \texttt{dir}(y') \land y'[\varnothing]$$

Success

$\exists y \cdot \texttt{resolve}(r, cwd, q/f, y) \land$

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

$\exists x \cdot \texttt{resolve}(r, cwd, q, x) \land \neg \texttt{dir}($



# Symbolic Execution

# Symbolic Execution

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

# Symbolic Execution

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

---

In progress

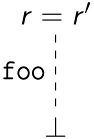
$r$

# Symbolic Execution

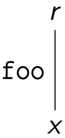
```
if [ -e foo ]; then
  rm foo
fi
```

Case 1

Success



In progress

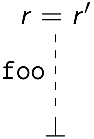


# Symbolic Execution

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

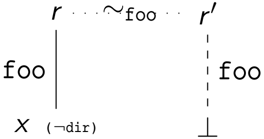
Case 1

Success



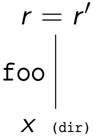
Case 2

Success



Case 3

Error

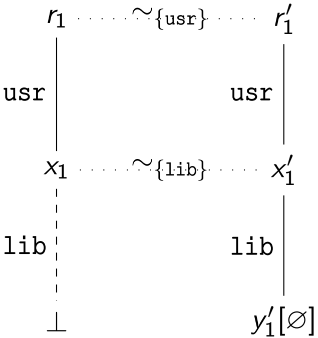


# Chaining Specifications

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

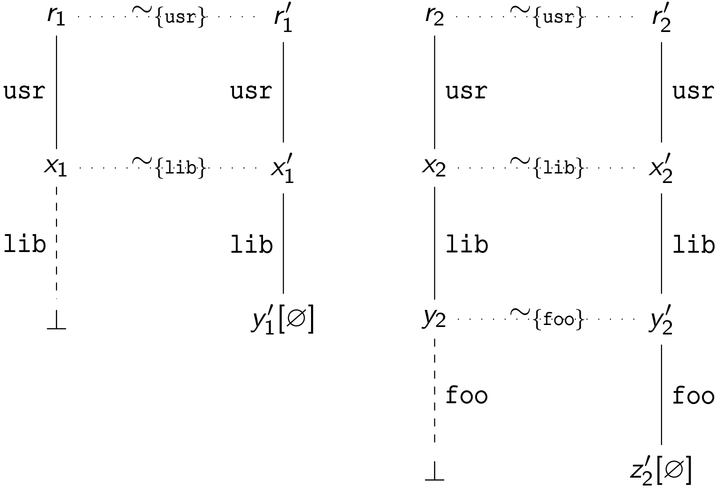
# Chaining Specifications

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



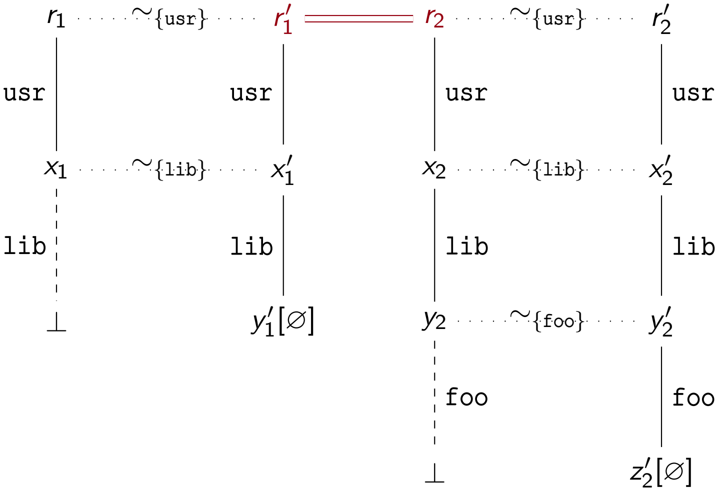
# Chaining Specifications

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



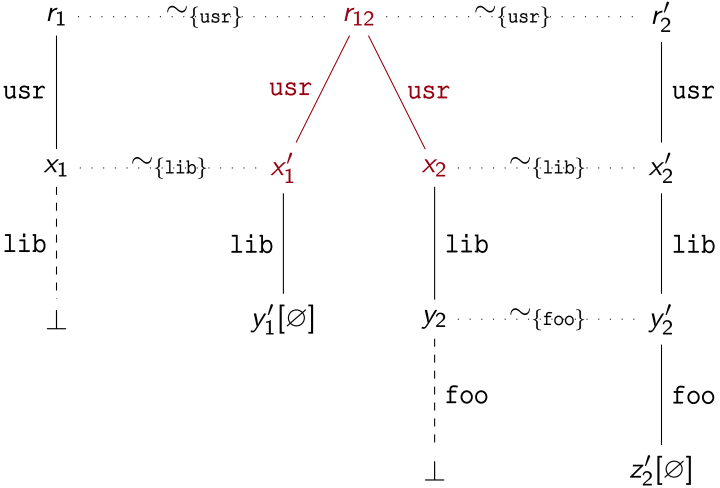
# Chaining Specifications

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


$r_1$ $\sim_{\{usr\}}$ $r'_1$
$r_2$ $\sim_{\{usr\}}$ $r'_2$
usr
usr
usr
usr
$x_1$ $\sim_{\{lib\}}$ $x'_1$
$x_2$ $\sim_{\{lib\}}$ $x'_2$
lib
lib
lib
lib
$\bot$
$y'_1[\varnothing]$
$y_2$ $\sim_{\{foo\}}$ $y'_2$
foo
foo
$\bot$
$z'_2[\varnothing]$

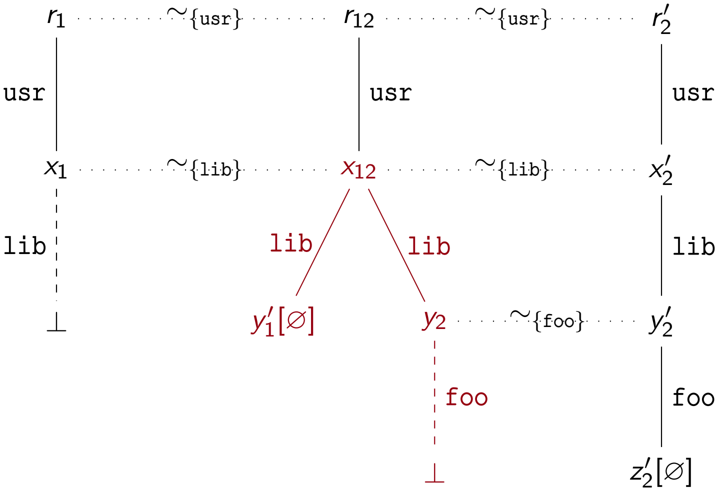
# Chaining Specifications

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

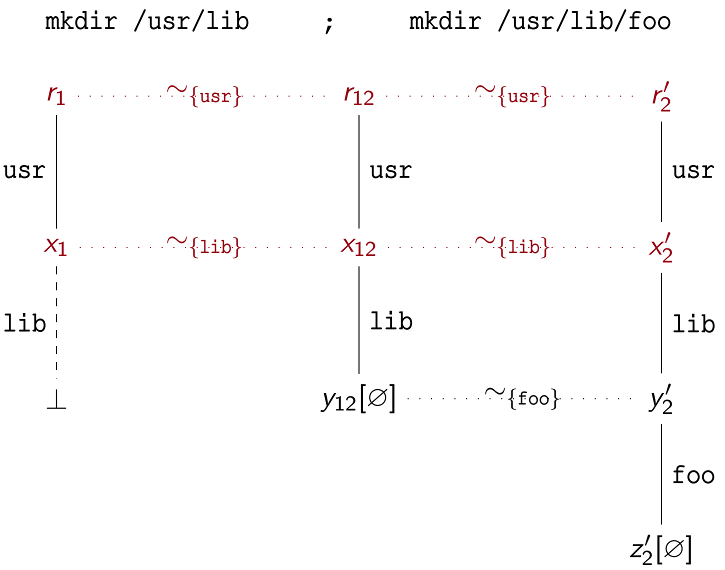


# Chaining Specifications

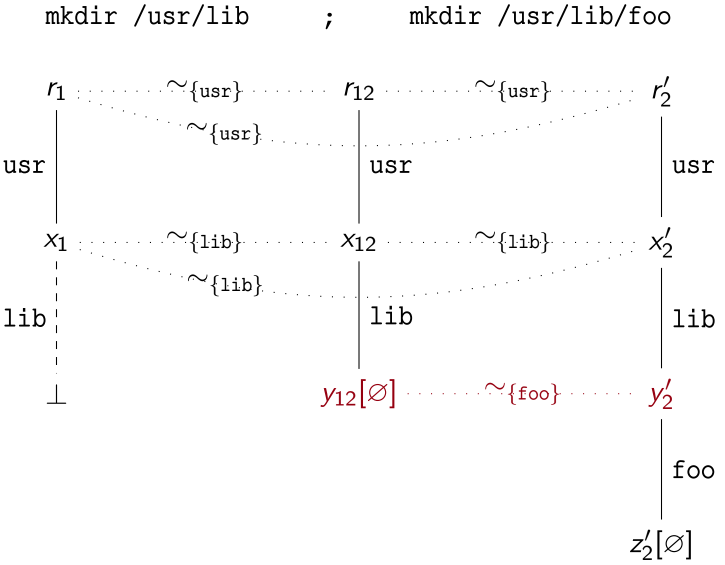
`mkdir /usr/lib ; mkdir /usr/lib/foo`



# Chaining Specifications

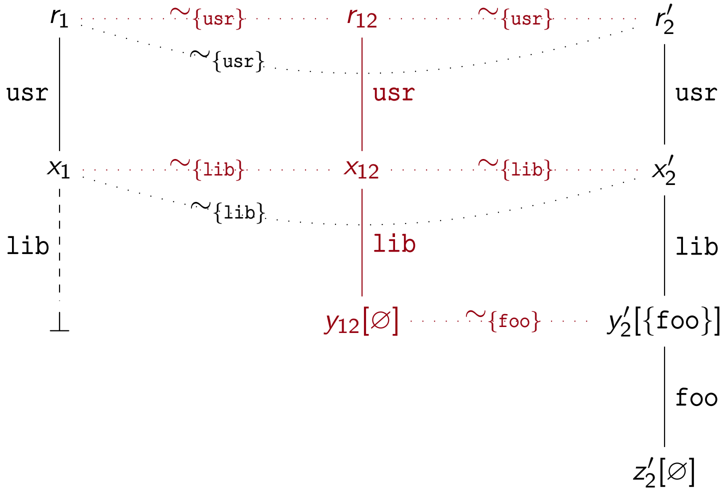


# Chaining Specifications


mkdir /usr/lib ; mkdir /usr/lib/foo
$r_1$
$\sim_{\{\texttt{usr}\}}$
$r_{12}$
$\sim_{\{\texttt{usr}\}}$
$r'_2$
$\sim_{\{\texttt{usr}\}}$
usr
usr
usr
$x_1$
$\sim_{\{\texttt{lib}\}}$
$x_{12}$
$\sim_{\{\texttt{lib}\}}$
$x'_2$
$\sim_{\{\texttt{lib}\}}$
lib
lib
lib
$\perp$
$y_{12}[\varnothing]$
$\sim_{\{\texttt{foo}\}}$
$y'_2$
foo
$z'_2[\varnothing]$

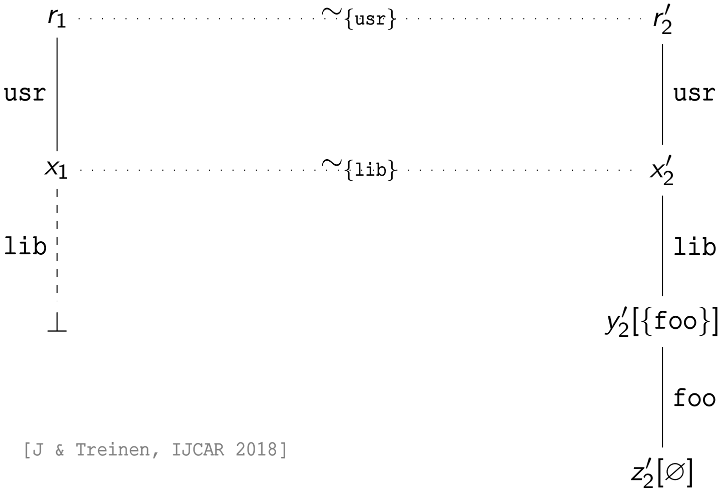
# Chaining Specifications

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



# Chaining Specifications

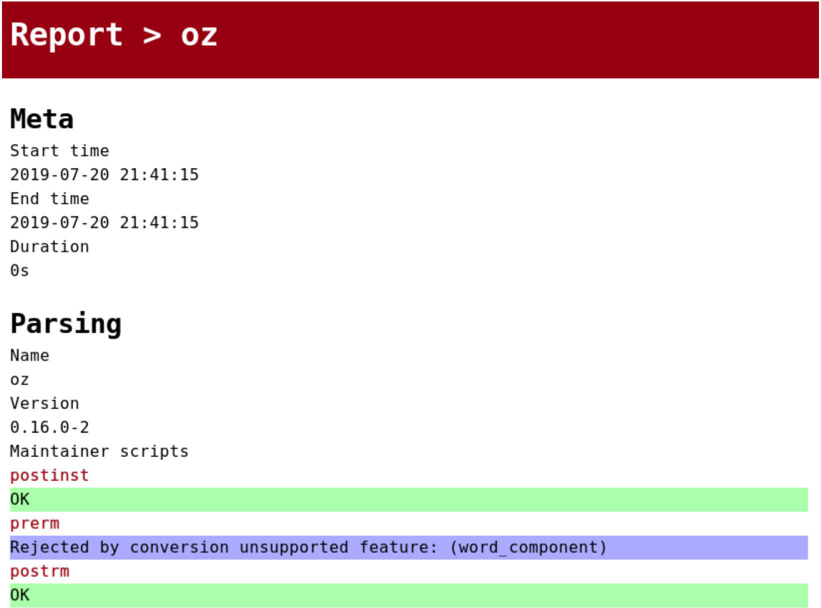
`mkdir /usr/lib ; mkdir /usr/lib/foo`



[J & Treinen, IJCAR 2018]

# Demo

# Package Report



## 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)

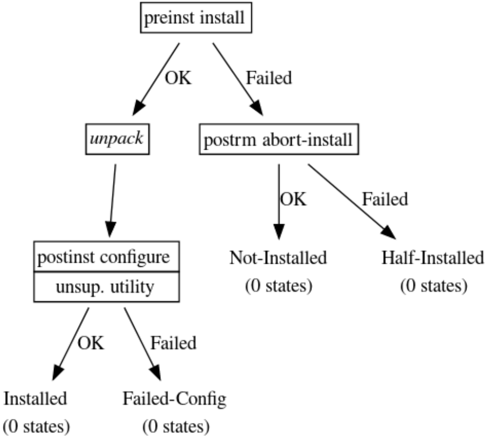
postrm

OK

# Installation Scenario

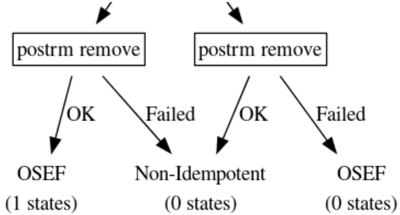
## Scenarii

### Installation



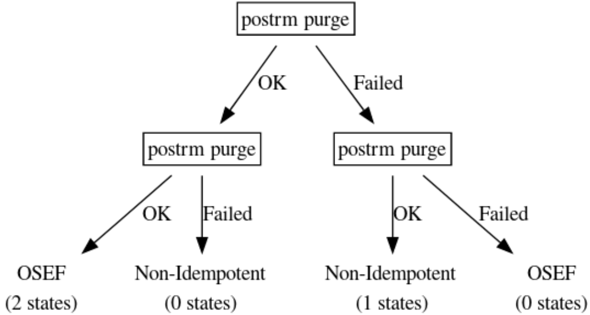
### Removal

# An Other Scenario



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

## Idempotency of postrm purge



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

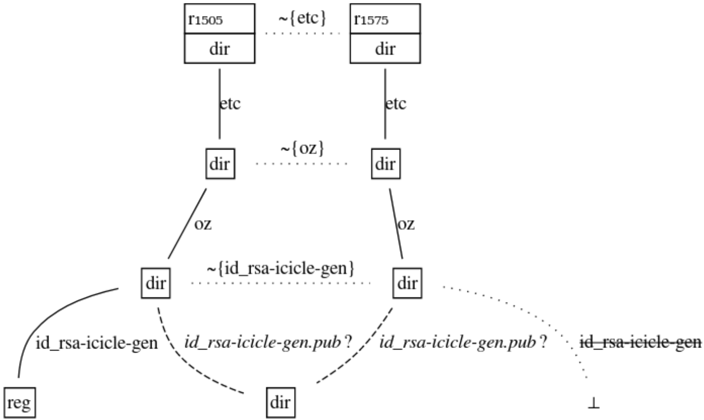
**OSEF**

0 1

**Non-Idempotent**

0

# An Execution Case

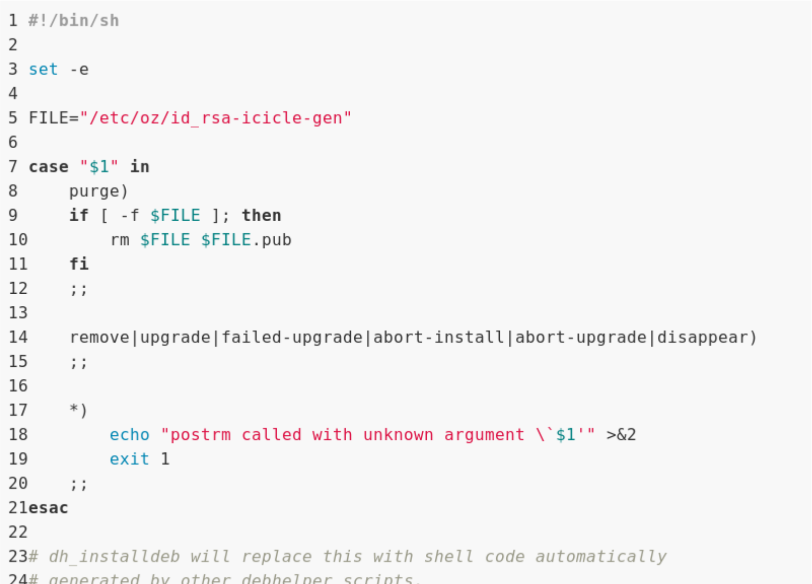


## log

```
[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
```

# The postrm Script

## Original Shell script



```
#!/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

# Conclusion

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

# 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/

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR
ANR

# 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!

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR
ANR

# 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;

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR
ANR

# 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!

PROJET FINANCÉ PAR L'ANR - PROJECT FUNDED BY THE ANR
ANR