

# Verification of Shell Scripts

## Performing File Hierarchy Transformations

Ph.D. Defence

Nicolas Jeannerod

Institut de Recherche en Informatique Fondamentale  
Université de Paris

March 30, 2021

# Introduction

# Software Installation

# Software Installation in Debian GNU/Linux

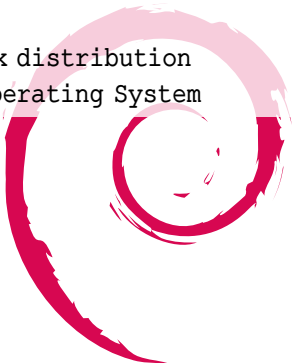
Debian GNU/Linux



# Software Installation in Debian GNU/Linux

Debian GNU/Linux

Linux distribution  
~ = Operating System



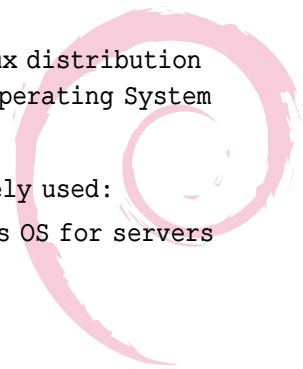
# Software Installation in Debian GNU/Linux

Debian GNU/Linux

Linux distribution  
~= Operating System

Widely used:

> as OS for servers



# Software Installation in Debian GNU/Linux

Debian GNU/Linux

Linux distribution  
~= Operating System

Widely used:

- > as OS for servers
- > as OS for desktop computers

# Software Installation in Debian GNU/Linux

Debian GNU/Linux

Linux distribution  
~= Operating System

Widely used:

- > as OS for servers
- > as OS for desktop computers
- > as basis for derived distributions - eg. Ubuntu



# Software Installation in Debian GNU/Linux

```
root@debian:~#
```

Debian GNU/Linux

Linux distribution  
~= Operating System

Widely used:

- > as OS for servers
- > as OS for desktop computers
- > as basis for derived distributions - eg. Ubuntu

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
```

Debian GNU/Linux

Linux distribution  
~= Operating System

Widely used:

- > as OS for servers
- > as OS for desktop computers
- > as basis for derived distributions - eg. Ubuntu

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed.
```

Debian GNU/Linux

Linux distribution  
~ = Operating System

Widely used:

- > as OS for servers
- > as OS for desktop computers
- > as basis for derived distributions - eg. Ubuntu

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed.
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 105.0-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
```

Debian GNU/Linux

Linux distribution  
~= Operating System

Widely used:

- > as OS for servers
- > as OS for desktop computers
- > as basis for derived distributions - eg. Ubuntu

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed.
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
```

Debian GNU/Linux

Linux distribution  
~ = Operating System

Widely used:

- > as OS for servers
- > as OS for desktop computers
- > as basis for derived distributions - eg. Ubuntu

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed.
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

Debian GNU/Linux

Linux distribution  
~ = Operating System

Widely used:

- > as OS for servers
- > as OS for desktop computers
- > as basis for derived distributions - eg. Ubuntu

# Software Packages and their Content

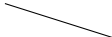


firefox\_74.0-1\_amd64.deb

# Software Packages and their Content



firefox\_74.0-1\_amd64.deb



control.tar.xz



data.tar.xz



# Software Packages and their Content



firefox\_74.0-1\_amd64.deb



control.tar.xz



data.tar.xz

/etc/firefox/firefox.js  
/usr/bin/firefox  
/usr/lib/firefox/application.ini  
/usr/lib/firefox/browser/blocklist.xml  
/usr/lib/firefox/browser/chrome  
/usr/lib/firefox/browser/crashreporter-override.ini  
/usr/lib/firefox/browser/defaults  
/usr/lib/firefox/browser/features/doh-rollout@mozilla.org  
/usr/lib/firefox/browser/features/formautofill@mozilla.org  
/usr/lib/firefox/browser/features/screenshots@mozilla.org

# Software Packages and their Content



firefox\_74.0-1\_amd64.deb



control.tar.xz



data.tar.xz



control



postinst



prerm

...

/etc/firefox/firefox.js

/usr/bin/firefox

/usr/lib/firefox/application.ini

/usr/lib/firefox/browser/blocklist.xml

/usr/lib/firefox/browser/chrome

/usr/lib/firefox/browser/crashreporter-override.ini

/usr/lib/firefox/browser/defaults

/usr/lib/firefox/browser/features/doh-rollout@mozilla.org

/usr/lib/firefox/browser/features/formautofill@mozilla.org

/usr/lib/firefox/browser/features/screenshots@mozilla.org

# Software Packages and their Content



firefox\_74.0-1\_a



control.tar.xz



control



postinst



prerm

...

Package: firefox

Version: 74.0-1

Architecture: amd64

...

Depends: libatk1.0-0 (>= 1.12.4), libc6 (>= 2.29), libcairo-gobject2 (1.10.0), libcairo2 (>= 1.10.0), ...

...

Description: Mozilla Firefox web browser

Firefox is a powerful, extensible web browser with support for modern web application technologies.

/usr/bin/firefox

/usr/lib/firefox/application.ini

/usr/lib/firefox/browser/blocklist.xml

/usr/lib/firefox/browser/chrome

/usr/lib/firefox/browser/crashreporter-override.ini

/usr/lib/firefox/browser/defaults

/usr/lib/firefox/browser/features/doh-rollout@mozilla.or

/usr/lib/firefox/browser/features/formautofill@mozilla.o

/usr/lib/firefox/browser/features/screenshots@mozilla.or

# Software Packages and their Content



firefox\_74.0-1\_a



control.tar.xz



control



postinst



prerm

...

Package: firefox

Version: 74.0-1

Architecture: amd64

...

Depends: libatk1.0-0 (>= 1.12.4), libc6 (>= 2.29), libcairo-gobject2 (1.10.0), libcairo2 (>= 1.10.0), ...

...

Description: Mozilla Firefox web browser

Firefox is a powerful, extensible web browser with support for modern

we #!/bin/sh -e

```
if [ "$1" = "remove" ] || [ "$1" = "deconfigure" ] ; then
    update-alternatives --remove x-www-browser /usr/bin/
    update-alternatives --remove gnome-www-browser /usr/
```

```
fi
```

```
if [ "$1" = "remove" ]; then
    rm -rf /usr/lib/firefox/updates
```

```
fi
```

# Software Installation in Debian GNU/Linux

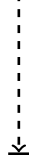
```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

User Request

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed.
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

User Request



Resolve

Dependencies

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

User Request



Resolve

Dependencies

Download

Package

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

User Request

✂

⋮

Resolve

⋮

Dependencies

⋮

⌵

✂

⋮

Download

⋮

Package

⋮

⌵

Run preinst



# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

User Request

✂

⋮

Resolve

⋮

Dependencies

⋮

⌵

✂

⋮

Download

⋮

Package

⋮

⌵

Run preinst

Unpack files

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

User Request

✂

⋮

Resolve

⋮

Dependencies

⋮

⌵

✂

⋮

Download

⋮

Package

⋮

⌵

Run preinst

Unpack files

Run postinst

# Software Installation in Debian GNU/Linux

```

root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#

```

```

User Request
┌
├─ Resolve
│  └─ Dependencies
└─
┌─ Download
│  └─ Package
└─
  Run preinst
  Unpack files
  Run postinst
┌
├─ Process
│  └─ Triggers
└─

```

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed.
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian unstable/main amd64 firefox amd64 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

User Request

✦

⋮ Resolve

⋮ Dependencies

⋮

⌵

✦

⋮ Download

⋮ Package

⋮

⌵

Run preinst

Unpack files

Run postinst

✦

⋮ Process

⋮ Triggers

⋮

⌵

Done

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed.
Need to get 0 B of space.
Get:1 http://deb.debian.org/debian/main amd64 firefox 74.0.1-1 [51.3 MB]
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

We are running Shell scripts

User Request

┌─┴─┐

┌─┴─┐ Resolve

┌─┴─┐ Dependencies

┌─┴─┐

┌─┴─┐ Download

┌─┴─┐ Package

┌─┴─┐

┌─┴─┐ Run preinst

┌─┴─┐ Unpack files

┌─┴─┐ Run postinst

┌─┴─┐

┌─┴─┐ Process

┌─┴─┐ Triggers

┌─┴─┐

Done

# Software Installation in Debian GNU/Linux

```
root@debian:~# apt install firefox
Reading package lists... Done
Building dependency tree
The following NEW packages will be installed:
  firefox
0 upgraded, 1 newly installed, 0 to remove and 0 not installed.
Need to get 51.3 MB of archives.
Get:1 http://deb.debian.org/debian/main amd64/firefox_74.0.1-1_amd64.deb 51.3 MB
Fetched 51.3 MB in 5s (9,569 kB/s)
Preparing to unpack .../firefox_74.0.1-1_amd64.deb ...
Unpacking firefox (74.0.1-1) ...
Setting up firefox (74.0.1-1) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for mime-support (3.64) ...
Processing triggers for gnome-menus (3.36.0-1) ...
root@debian:~#
```

We are running Shell scripts  
with full privileges.

User Request

✦  
⋮  
⋮ Resolve  
⋮ Dependencies  
⋮  
⋮  
✦  
⋮ Download  
⋮ Package  
⋮  
⋮  
✦ Run preinst  
Unpack files  
✦ Run postinst  
⋮  
⋮ Process  
⋮ Triggers  
⋮  
✦ Done

## What Could Possibly Go Wrong?

**From:** "Aaron M. Ucko" <ucko@debian.org>  
**To:** Debian Bug Tracking System <submit@bugs.debian.org>  
**Subject:** cmigrep: broken emacs-environment script  
**Date:** Fri, 29 Jun 2007 20:27:06 -0400

Package: cmigrep

Version: 1.3-1

Severity: critical

Justification: breaks unrelated software

cmigrep's emacs-environment script is overzealous; specifically, it inappropriately attempts to compile all .el files in /usr/share/emacs/site-lisp even if they don't work with the current emacs flavor (for instance, remembrance-agent's remem.el

## Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.



## Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:

## Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:
  - > make other software unusable,

## Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:
  - > make other software unusable,
  - > make the whole computer unusable,

# Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:
  - > make other software unusable,
  - > make the whole computer unusable,
  - > remove your personal files.

# Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:
  - > make other software unusable,
  - > make the whole computer unusable,
  - > remove your personal files.
- > Why it is hard? Because we manipulate:

## Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:
  - > make other software unusable,
  - > make the whole computer unusable,
  - > remove your personal files.
- > Why it is hard? Because we manipulate:
  - > POSIX Shell scripts with:

# Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:
  - > make other software unusable,
  - > make the whole computer unusable,
  - > remove your personal files.
- > Why it is hard? Because we manipulate:
  - > POSIX Shell scripts with:
    - > treacherous syntax,

# Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:
  - > make other software unusable,
  - > make the whole computer unusable,
  - > remove your personal files.
- > Why it is hard? Because we manipulate:
  - > POSIX Shell scripts with:
    - > treacherous syntax,
    - > unusual, complex semantics;



# Problem of this Ph.D. Thesis

- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:
  - > make other software unusable,
  - > make the whole computer unusable,
  - > remove your personal files.
- > Why it is hard? Because we manipulate:
  - > POSIX Shell scripts with:
    - > treacherous syntax,
    - > unusual, complex semantics;
  - > Unix filesystems: complex tree-like data structures;

# Problem of this Ph.D. Thesis

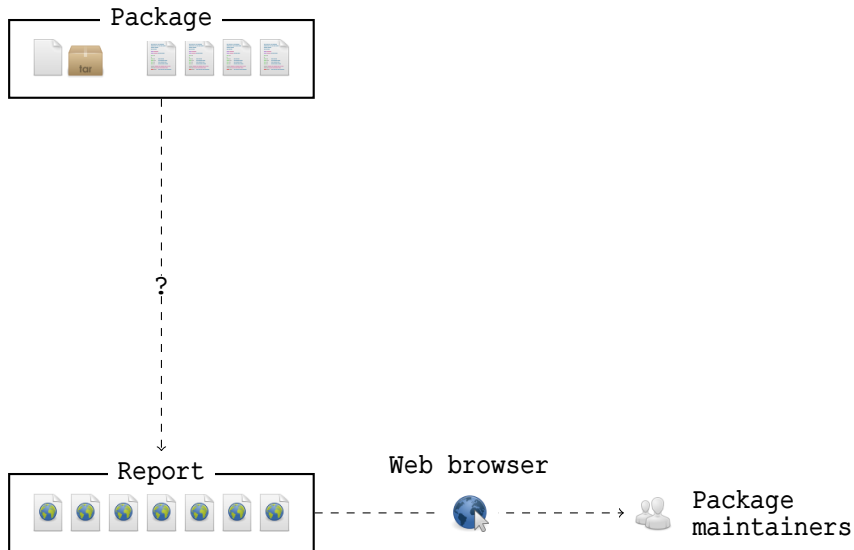
- > Goal: applying formal methods to Shell scripts and to the quality assessment of Debian Packages in particular.
- > Goal (reformulated): making sure that installing/updating/removing software does not:
  - > make other software unusable,
  - > make the whole computer unusable,
  - > remove your personal files.
- > Why it is hard? Because we manipulate:
  - > POSIX Shell scripts with:
    - > treacherous syntax,
    - > unusual, complex semantics;
  - > Unix filesystems: complex tree-like data structures;
  - > and Unix utilities: transformations of such filesystems.

# Battle Plan

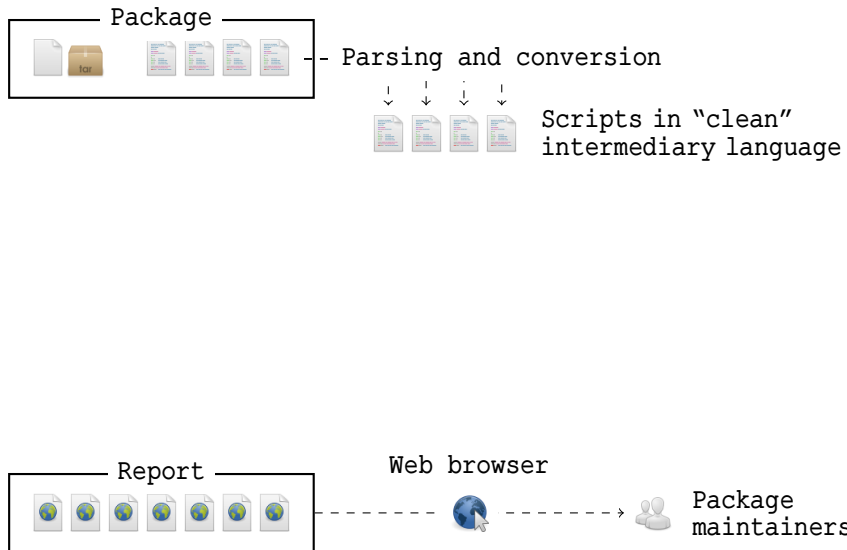
## Package



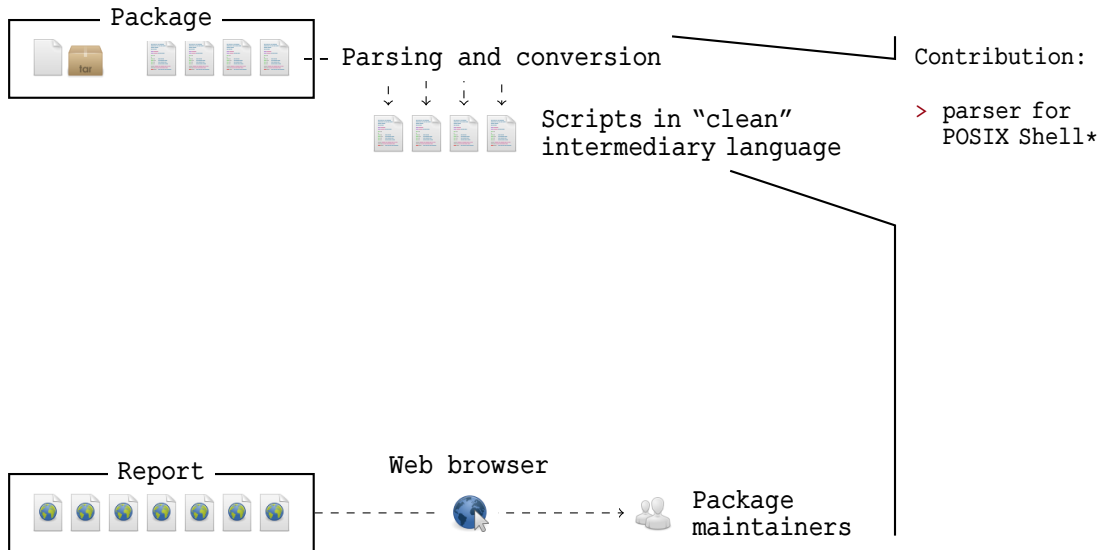
# Battle Plan



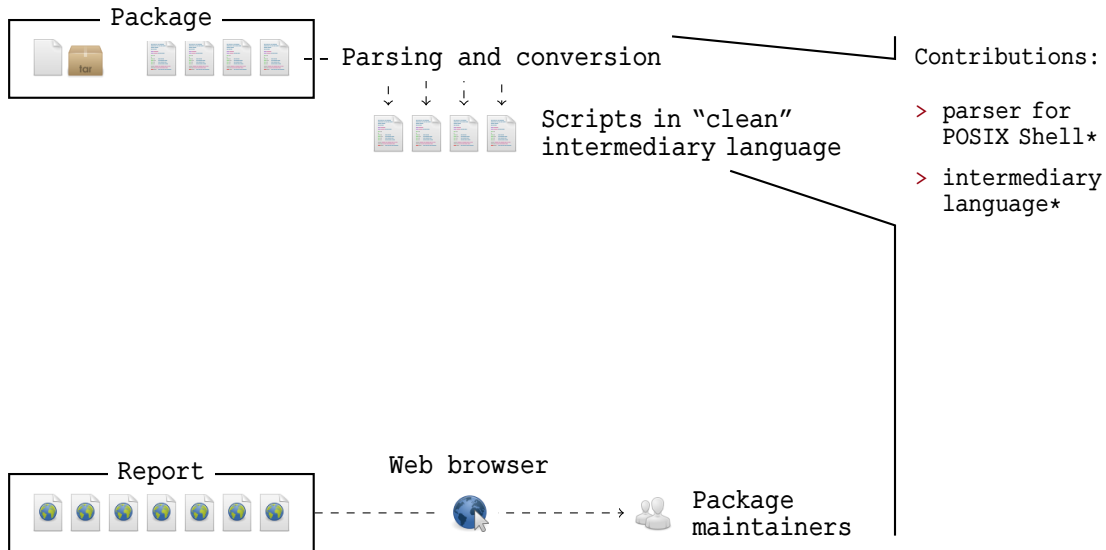
# Battle Plan



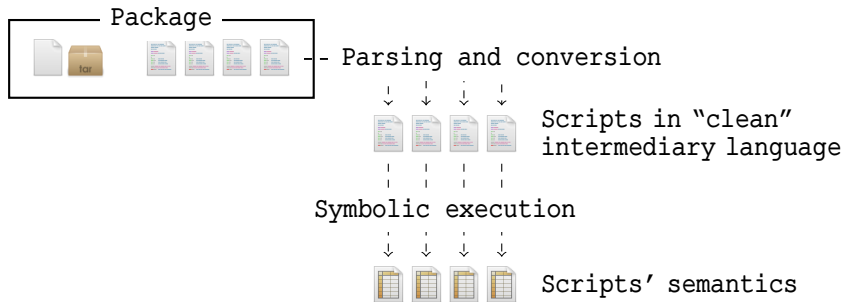
# Battle Plan



# Battle Plan

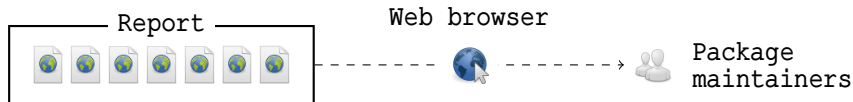


# Battle Plan



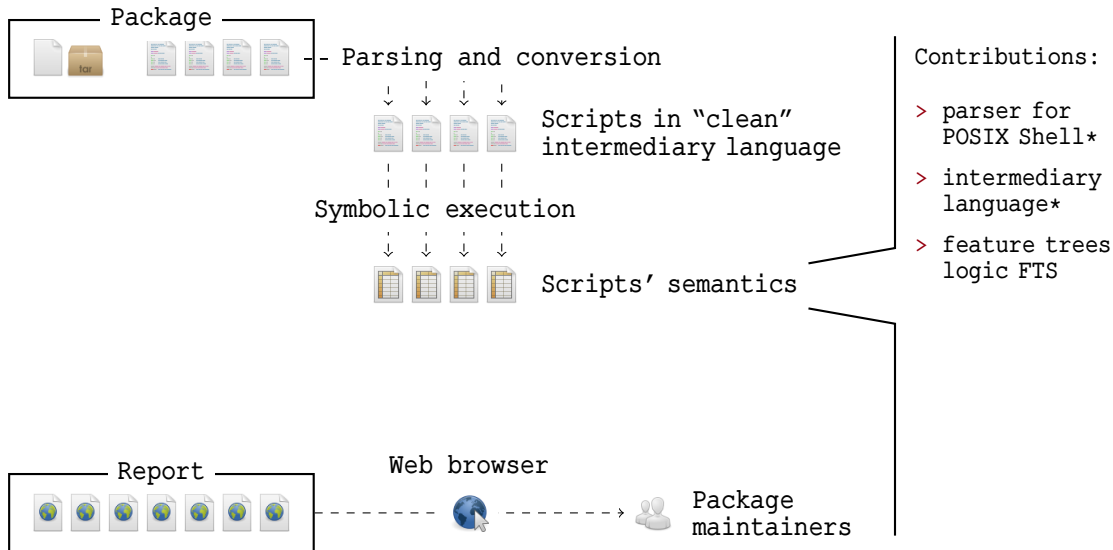
## Contributions:

- > parser for POSIX Shell\*
- > intermediary language\*

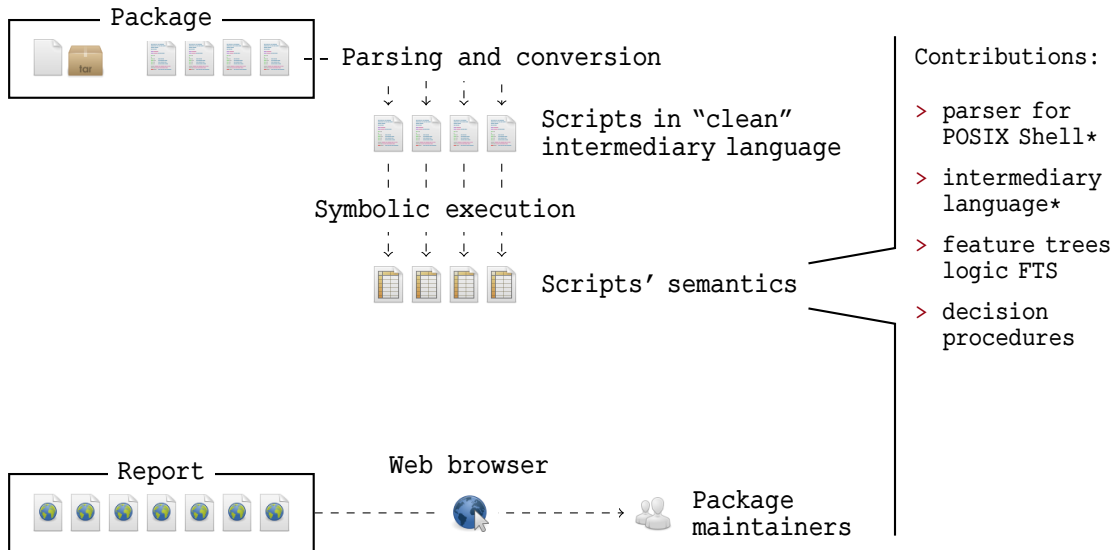




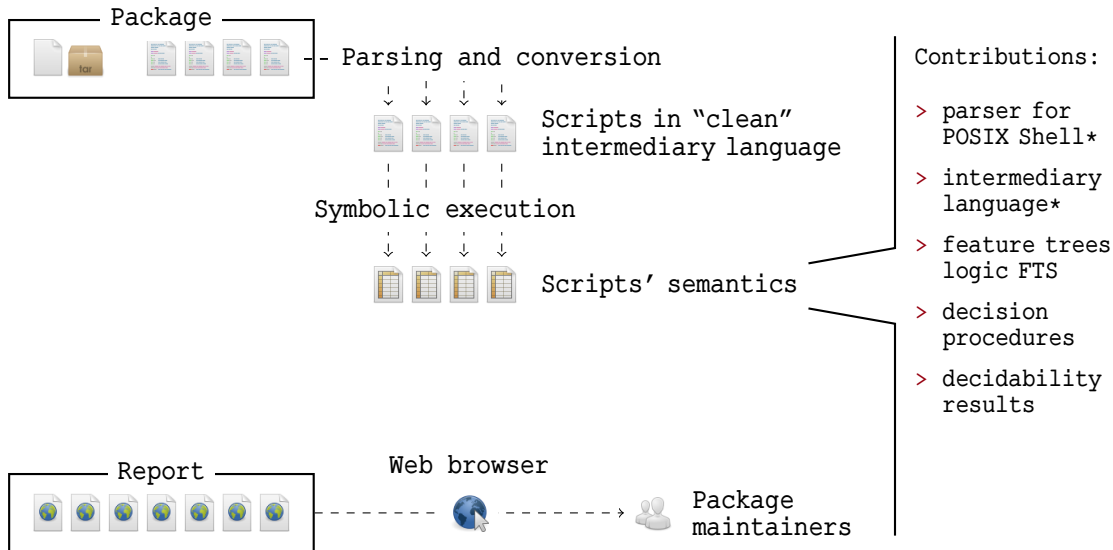
# Battle Plan



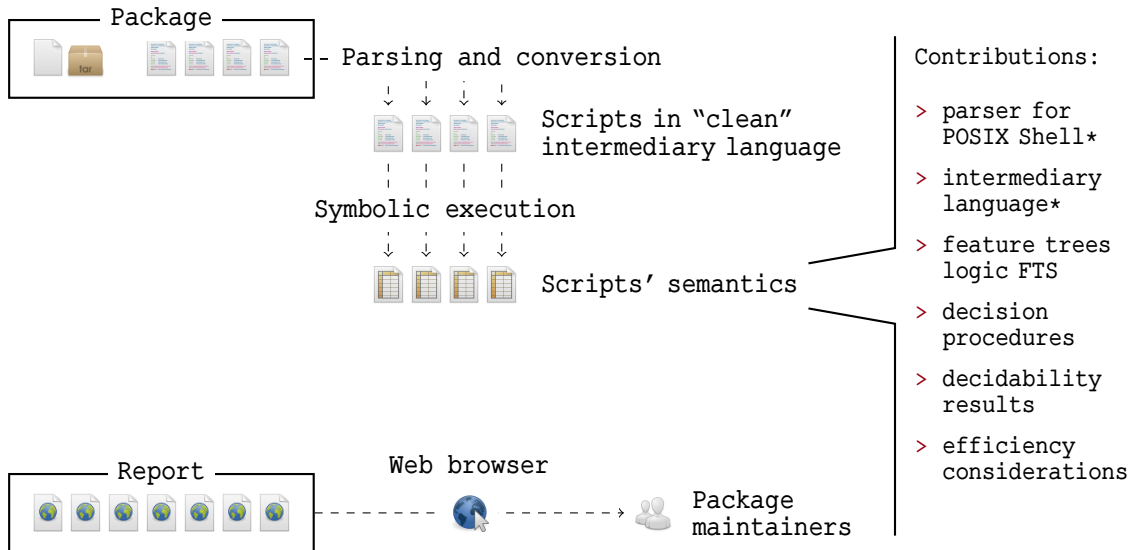
# Battle Plan



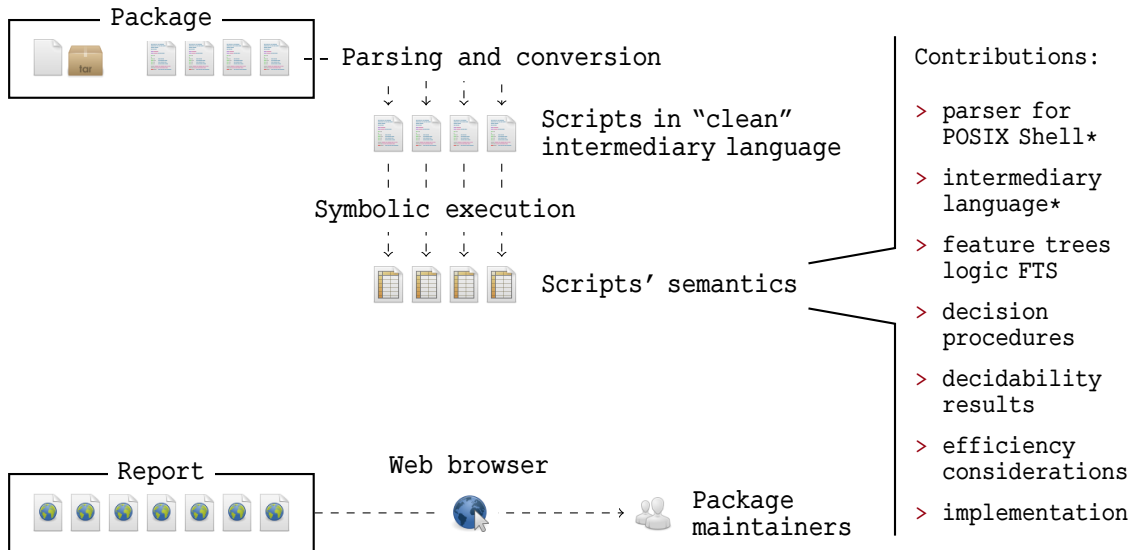
# Battle Plan



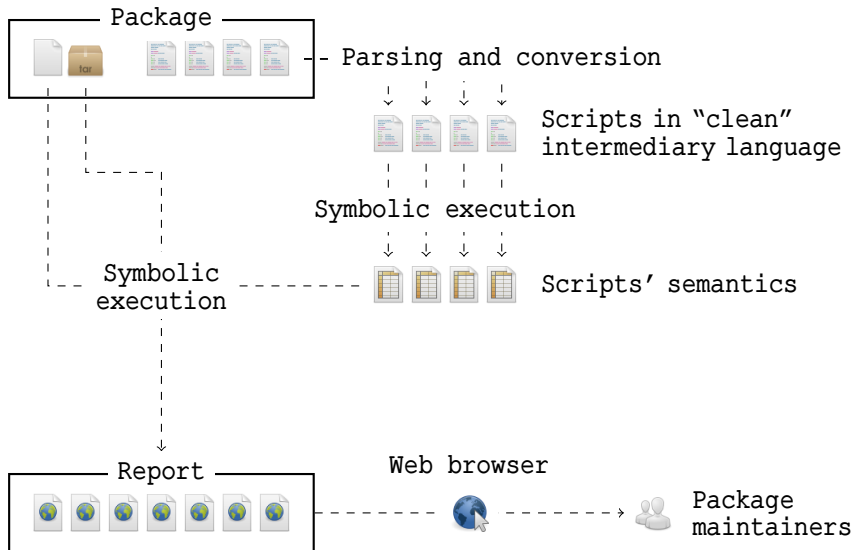
# Battle Plan



# Battle Plan



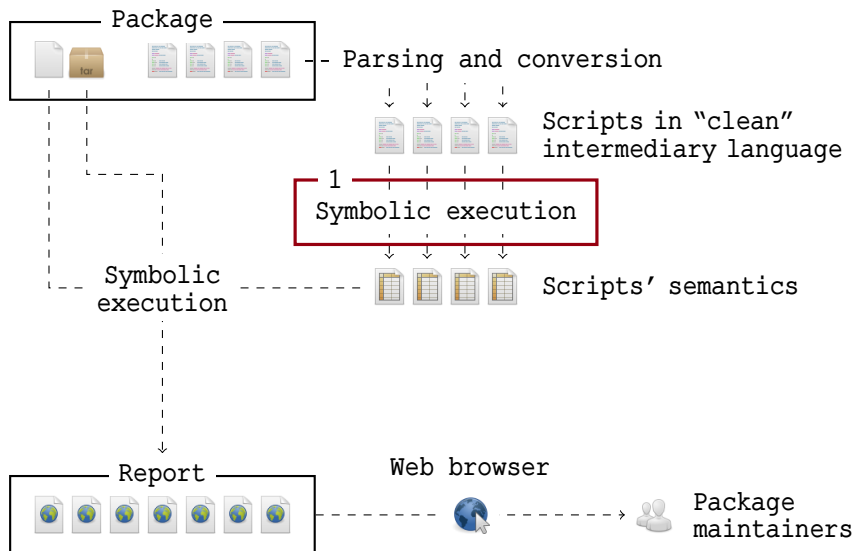
# Battle Plan



## Contributions:

- > parser for POSIX Shell\*
- > intermediary language\*
- > feature trees logic FTS
- > decision procedures
- > decidability results
- > efficiency considerations
- > implementation

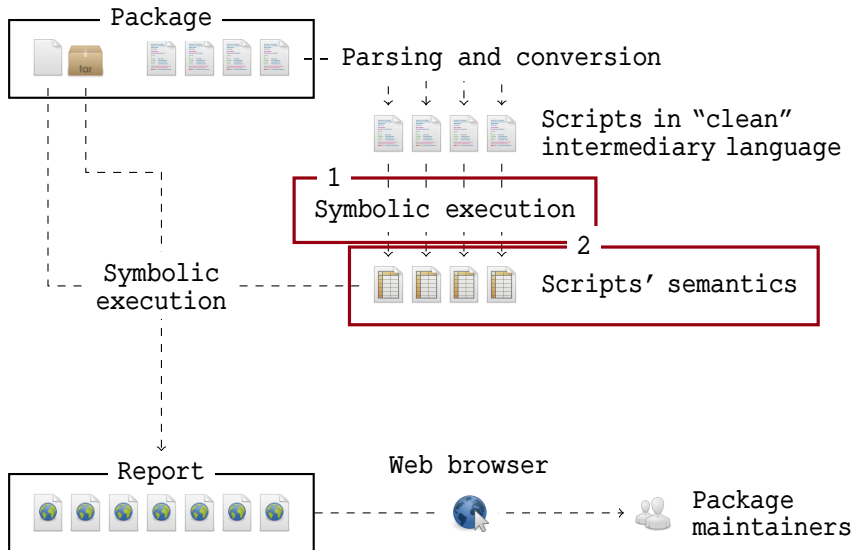
# Battle Plan



## Contributions:

- > parser for POSIX Shell\*
- > intermediary language\*
- > feature trees logic FTS
- > decision procedures
- > decidability results
- > efficiency considerations
- > implementation

# Battle Plan

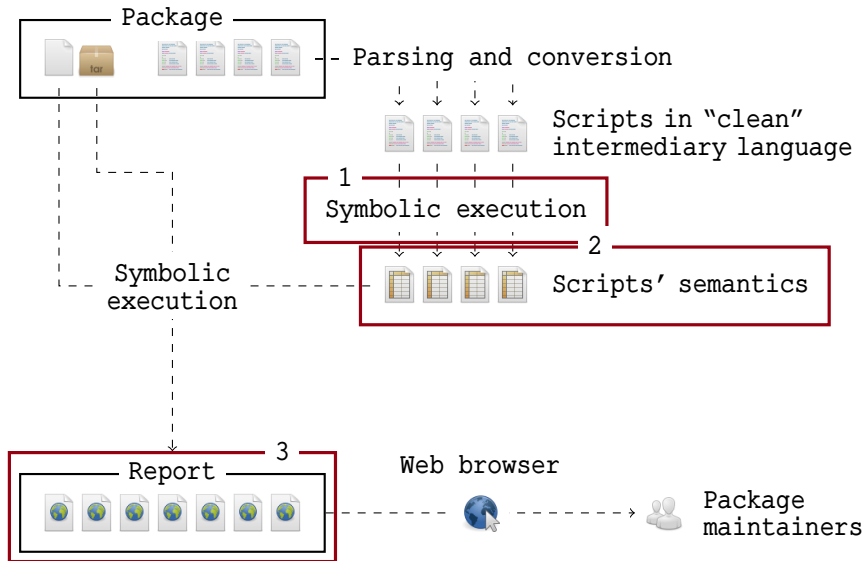


## Contributions:

- > parser for POSIX Shell\*
- > intermediary language\*
- > feature trees logic FTS
- > decision procedures
- > decidability results
- > efficiency considerations
- > implementation



# Battle Plan



## Contributions:

- > parser for POSIX Shell\*
- > intermediary language\*
- > feature trees logic FTS
- > decision procedures
- > decidability results
- > efficiency considerations
- > implementation

## Symbolic Execution

# Example Shell Script

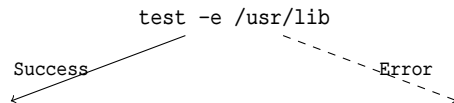
```
1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi
```

# Example Shell Script and its Execution Traces

```
test -e /usr/lib
```

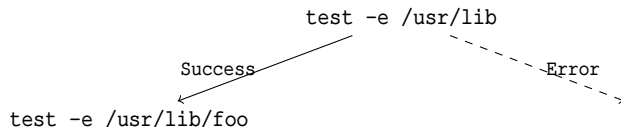
```
1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi
```

# Example Shell Script and its Execution Traces



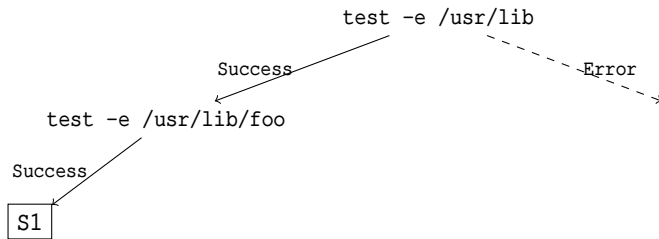
```
1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi
```

# Example Shell Script and its Execution Traces



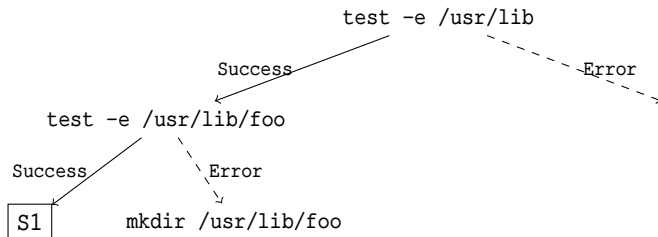
```
1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi
```

## Example Shell Script and its Execution Traces



```
1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi
```

# Example Shell Script and its Execution Traces

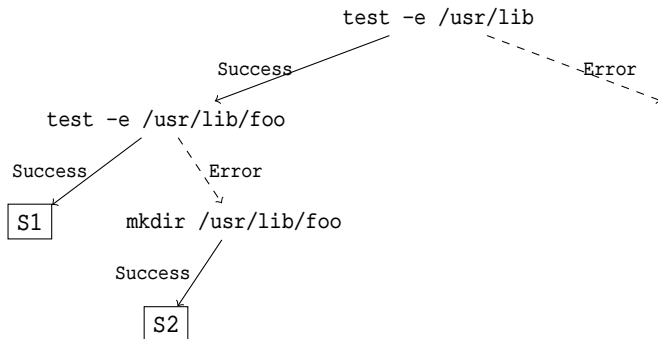


```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi
  
```



# Example Shell Script and its Execution Traces

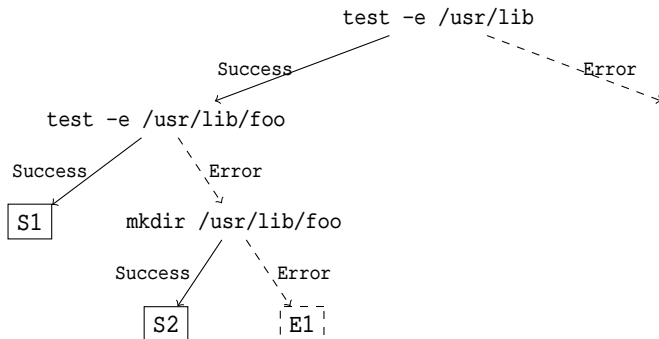


```

1  if ! test -e /usr/lib; then
2    mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5    mkdir /usr/lib/foo
6  fi

```

# Example Shell Script and its Execution Traces

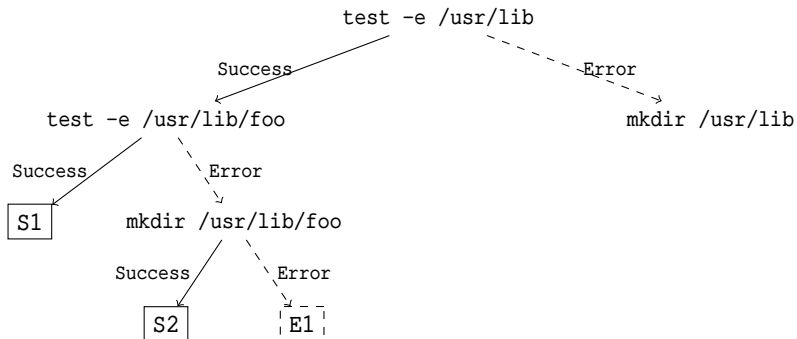


```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi

```

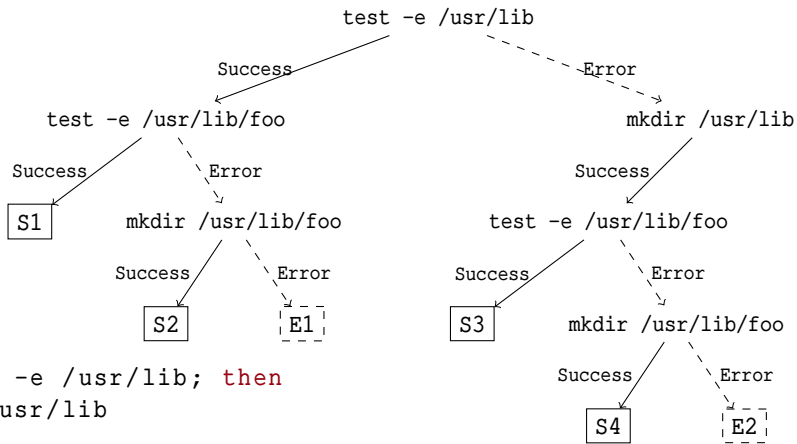
# Example Shell Script and its Execution Traces



```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi
  
```

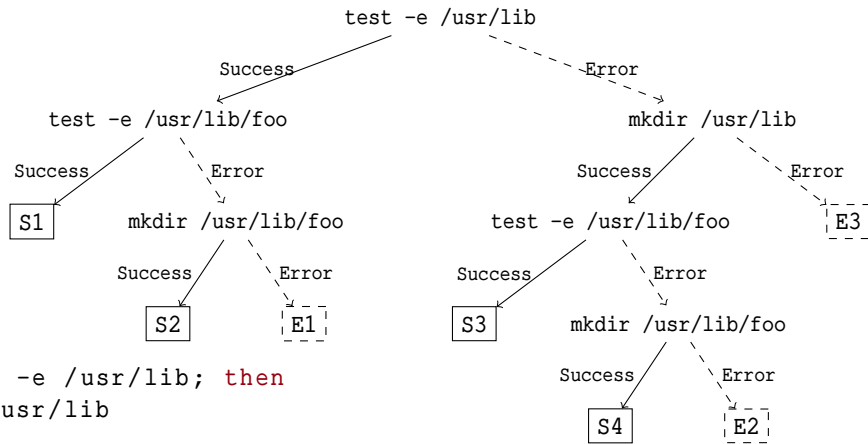
# Example Shell Script and its Execution Traces



```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi
  
```

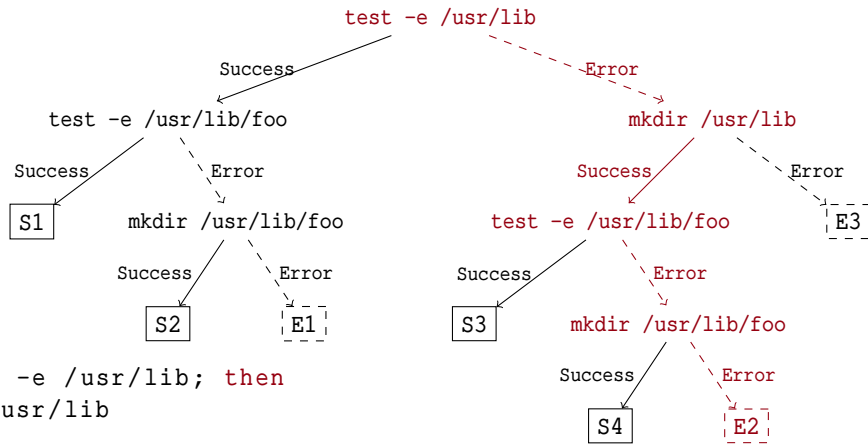
# Example Shell Script and its Execution Traces



```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi
  
```

# Example Shell Script and its Execution Traces



```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi

```

# Backend Requirements

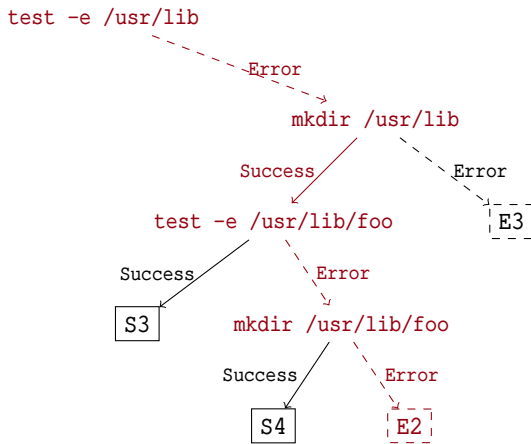
Logic Requirements:

- > Express tree relations

```

1  if ! test -e /usr/lib; then
2    mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5    mkdir /usr/lib/foo
6  fi

```



# Backend Requirements

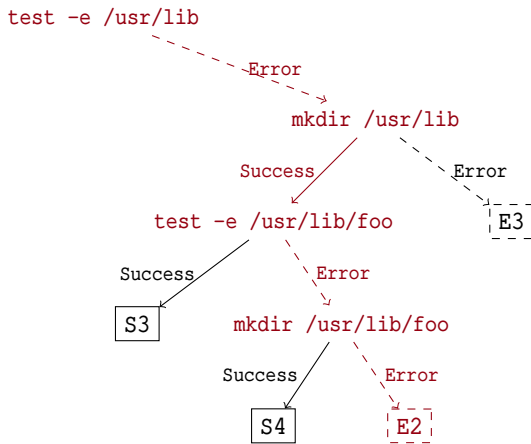
Logic Requirements:

- > Express tree relations
- > Enough expressivity to specify Unix utilities

```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi

```





# Backend Requirements

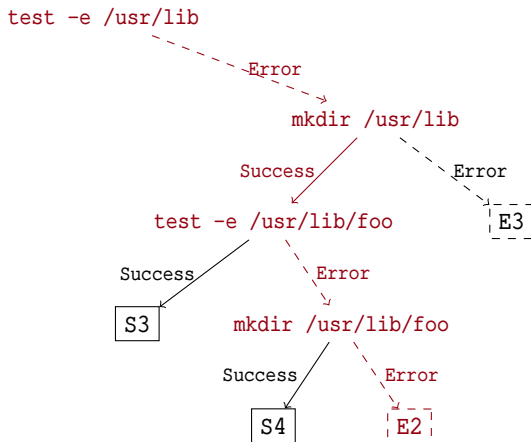
Logic Requirements:

- > Express tree relations
- > Enough expressivity to specify Unix utilities
- > Express the composition of tree relations

```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi

```



# Backend Requirements

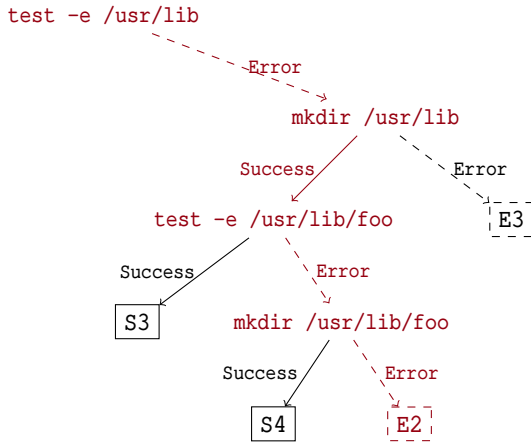
Logic Requirements:

- > Express tree relations
- > Enough expressivity to specify Unix utilities
- > Express the composition of tree relations
- > Detection of impossible cases

```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi

```



# Backend Requirements

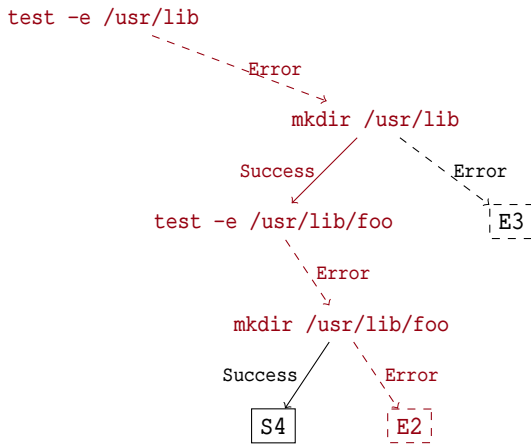
Logic Requirements:

- > Express tree relations
- > Enough expressivity to specify Unix utilities
- > Express the composition of tree relations
- > Detection of impossible cases

```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi

```



# Backend Requirements

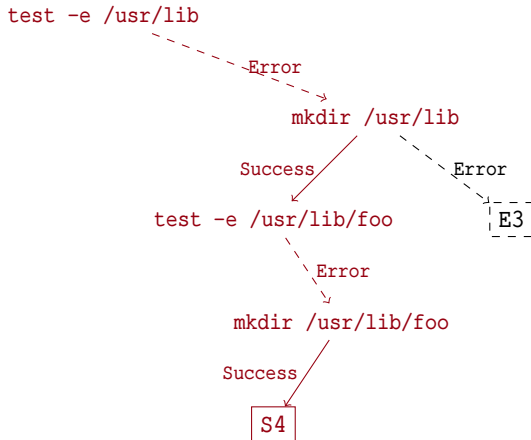
Logic Requirements:

- > Express tree relations
- > Enough expressivity to specify Unix utilities
- > Express the composition of tree relations
- > Detection of impossible cases

```

1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi

```



# Backend Requirements

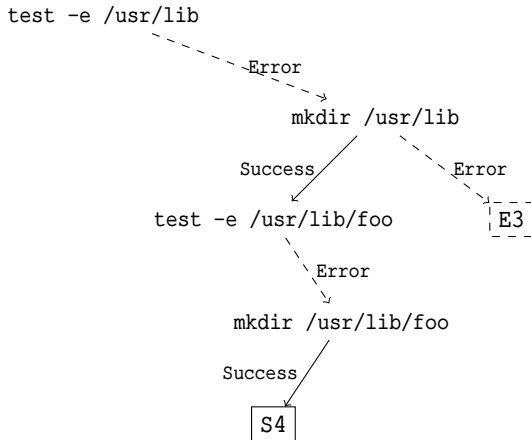
Logic Requirements:

- > Express tree relations
- > Enough expressivity to specify Unix utilities
- > Express the composition of tree relations
- > Detection of impossible cases
- > Incrementality

```

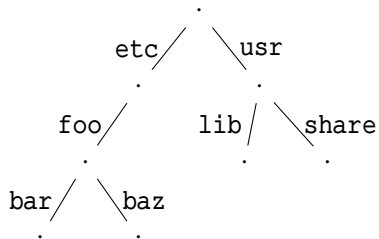
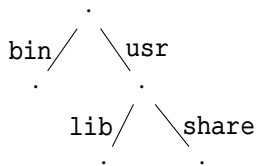
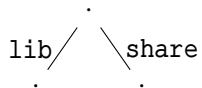
1  if ! test -e /usr/lib; then
2      mkdir /usr/lib
3  fi
4  if ! test -e /usr/lib/foo; then
5      mkdir /usr/lib/foo
6  fi

```

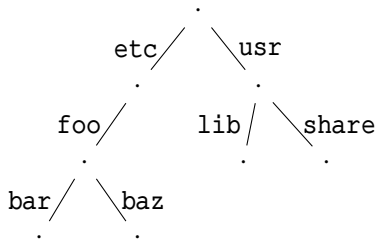
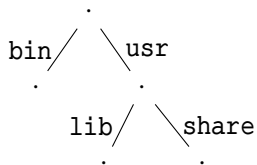
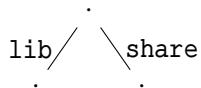


## Syntax & Semantics of FTS

# Feature Trees



# Feature Trees



Set of feature trees  
(inductively defined)

$$\mathcal{FT} = \mathcal{F} \rightsquigarrow \mathcal{FT}$$

Set of features  
(legal file names)

Partial functions  
with finite domain



# Syntax & Semantics of FTS

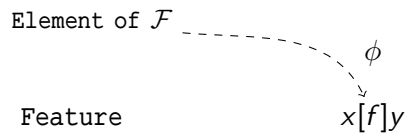
## Syntax &amp; Semantics of FTS

 $\phi$ 

Feature

 $x[f]y$

## Syntax &amp; Semantics of FTS



## Syntax &amp; Semantics of FTS

Element of  $\mathcal{F}$  $\phi$ 

Feature

 $x[f]y$  $\rho$  satisfies  $\phi$  if

$$\rho(x)(f) = \rho(y)$$

# Syntax & Semantics of FTS

Element of  $\mathcal{F}$

$\phi$

$\rho$  satisfies  $\phi$  if

Feature

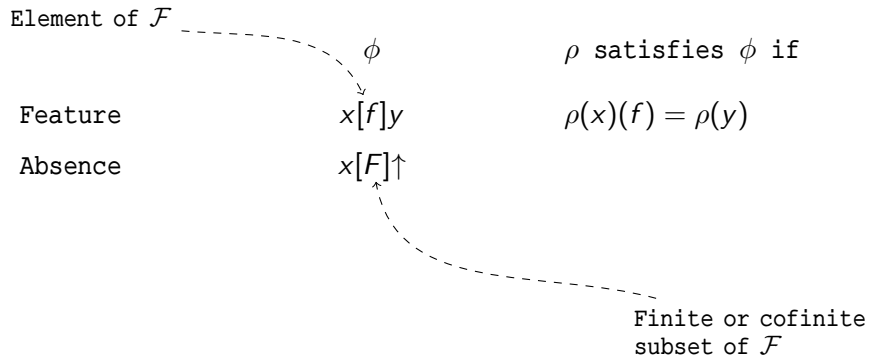
$x[f]y$

$\rho(x)(f) = \rho(y)$

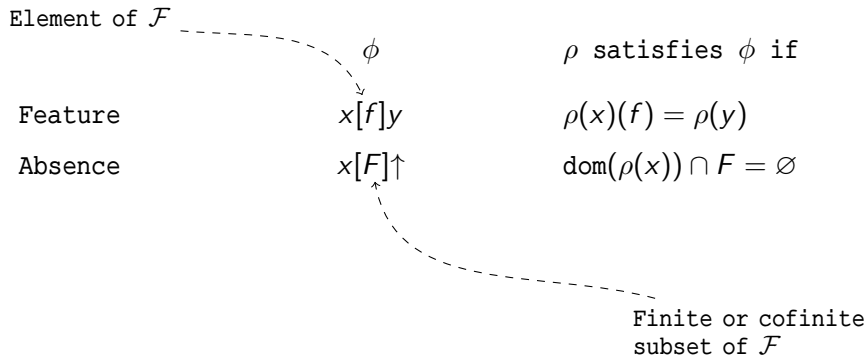
Absence

$x[F]\uparrow$

# Syntax & Semantics of FTS



# Syntax & Semantics of FTS



# Syntax & Semantics of FTS

Element of  $\mathcal{F}$

$\phi$

$\rho$  satisfies  $\phi$  if

Feature

$x[f]y$

$\rho(x)(f) = \rho(y)$

Absence

$x[F]\uparrow$

$\text{dom}(\rho(x)) \cap F = \emptyset$

Similarity

$x =_F y$

Finite or cofinite  
subset of  $\mathcal{F}$



# Syntax & Semantics of FTS

Element of $\mathcal{F}$	$\phi$	$\rho$ satisfies $\phi$ if
Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$

Finite or cofinite subset of  $\mathcal{F}$

# Syntax & Semantics of FTS

Element of $\mathcal{F}$	$\phi$	$\rho$ satisfies $\phi$ if
Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$
Negation	$\neg\phi$	Finite or cofinite subset of $\mathcal{F}$
Conjunction	$\phi \wedge \psi$	
Disjunction	$\phi \vee \psi$	
Exist. Quant.	$\exists x \cdot \phi$	
Univ. Quant.	$\forall x \cdot \phi$	

# Syntax & Semantics of FTS

Element of $\mathcal{F}$	$\phi$	$\rho$ satisfies $\phi$ if
Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$
Negation	$\neg\phi$	Finite or cofinite subset of $\mathcal{F}$
Conjunction	$\phi \wedge \psi$	
Disjunction	$\phi \vee \psi$	
Exist. Quant.	$\exists x \cdot \phi$	Quantification over variables only
Univ. Quant.	$\forall x \cdot \phi$	

## Syntax &amp; Semantics of FTS

	Element of $\mathcal{F}$	$\phi$	$\rho$ satisfies $\phi$ if
FT	Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
	Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
	Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$
	Negation	$\neg\phi$	Finite or cofinite subset of $\mathcal{F}$
	Conjunction	$\phi \wedge \psi$	
	Disjunction	$\phi \vee \psi$	
	Exist. Quant.	$\exists x \cdot \phi$	Quantification over variables only
	Univ. Quant.	$\forall x \cdot \phi$	

# Syntax & Semantics of FTS

	Element of $\mathcal{F}$	$\phi$	$\rho$ satisfies $\phi$ if
CFT	Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
	Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
	Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$
	Negation	$\neg\phi$	Finite or cofinite subset of $\mathcal{F}$
	Conjunction	$\phi \wedge \psi$	
	Disjunction	$\phi \vee \psi$	
	Exist. Quant.	$\exists x \cdot \phi$	Quantification over variables only
	Univ. Quant.	$\forall x \cdot \phi$	

# Syntax & Semantics of FTS

	$\phi$	$\rho$ satisfies $\phi$ if
Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$

$$\begin{array}{c} x \\ f \mid \\ y \end{array}$$

Feature

## Syntax &amp; Semantics of FTS

	$\phi$	$\rho$ satisfies $\phi$ if
Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$

$$\begin{array}{c} x \\ f \mid \\ y \end{array}$$

Feature

$$\begin{array}{c} x \\ \not f \vdots \\ \perp \end{array}$$

Absence

## Syntax &amp; Semantics of FTS

	$\phi$	$\rho$ satisfies $\phi$ if
Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$

$$\begin{array}{c} x \\ f \mid \\ y \end{array}$$

Feature

$$\begin{array}{c} x \\ \not f \vdots \\ \perp \end{array} \quad x[F]\uparrow$$

Absence



## Syntax &amp; Semantics of FTS

	$\phi$	$\rho$ satisfies $\phi$ if
Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$

$$\begin{array}{c} x \\ f \mid \\ y \end{array}$$

Feature

$$\begin{array}{c} x \\ \vdots \\ \perp \end{array}$$

Absence

 $x[F]\uparrow$ 
 $x \dots =_F \dots y$ 

Similarity

# Syntax & Semantics of FTS

	$\phi$	$\rho$ satisfies $\phi$ if
Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$

$$x \stackrel{\text{def}}{=} y$$

Equality

## Syntax &amp; Semantics of FTS

	$\phi$	$\rho$ satisfies $\phi$ if
Feature	$x[f]y$	$\rho(x)(f) = \rho(y)$
Absence	$x[F]\uparrow$	$\text{dom}(\rho(x)) \cap F = \emptyset$
Similarity	$x =_F y$	$\rho(x) =_F \rho(y)$

$$x \dots \overset{=}{\star} \dots y$$

Equality

$$\begin{array}{ccc}
 x & \dots \overset{=}{\mathbb{C}\{f\}} \dots & y \\
 \vdots & & \vdots \\
 f & & f \\
 \vdots & & \vdots \\
 \perp & & z
 \end{array}$$

Update

## Example Specification

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

# Example Specification

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

```

  r
usr |
   $\exists x$ 
lib |
   $\exists y$ 
foo |
   $\perp$ 

```

Success case

# Example Specification

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

	$r \dots \equiv \mathcal{C}\{\text{usr}\} \dots r'$
usr	usr
	$\exists x \dots \equiv \mathcal{C}\{\text{lib}\} \dots \exists x'$
lib	lib
	$\exists y \dots \equiv \mathcal{C}\{\text{foo}\} \dots \exists y'$
foo	foo
$\perp$	$\exists z'[\star]\uparrow$

Success case

# Example Specification

mkdir /usr/lib/foo

$$\begin{array}{ccc}
 r & \dots\dots\dots \models \mathcal{C}\{\text{usr}\} & \dots\dots\dots r' \\
 \text{usr} \mid & & \mid \text{usr} \\
 \exists x & \dots\dots\dots \models \mathcal{C}\{\text{lib}\} & \dots\dots\dots \exists x' \\
 \text{lib} \mid & & \mid \text{lib} \\
 \exists y & \dots\dots\dots \models \mathcal{C}\{\text{foo}\} & \dots\dots\dots \exists y' \\
 \text{foo} \mid & & \mid \text{foo} \\
 \perp & & \exists z'[\star]\uparrow
 \end{array}$$

Success case

$$\begin{array}{c}
 r =_{\star} r' \\
 \mid \text{usr} \\
 \perp
 \end{array}$$

Error case

# Example Specification

mkdir /usr/lib/foo

	$r \dots \equiv \mathcal{C}\{\text{usr}\} \dots r'$	
usr		usr
	$\exists x \dots \equiv \mathcal{C}\{\text{lib}\} \dots \exists x'$	
lib		lib
	$\exists y \dots \equiv \mathcal{C}\{\text{foo}\} \dots \exists y'$	
foo		foo
	$\perp$	$\exists z'[\star]\uparrow$

Success case

$r =_\star r'$	$r =_\star r'$
<del>usr</del>	usr
$\perp$	$\exists x$
	<del>lib</del>
	$\perp$

Error cases



# Example Specification

mkdir /usr/lib/foo

$r$	$\dots \equiv \mathcal{C}\{\text{usr}\} \dots$	$r'$
usr		usr
	$\exists x \dots \equiv \mathcal{C}\{\text{lib}\} \dots$	$\exists x'$
lib		lib
	$\exists y \dots \equiv \mathcal{C}\{\text{foo}\} \dots$	$\exists y'$
foo		foo
	$\perp$	$\exists z'[\star]\uparrow$

Success case

$r =_\star r'$
<del>usr</del>
$\perp$

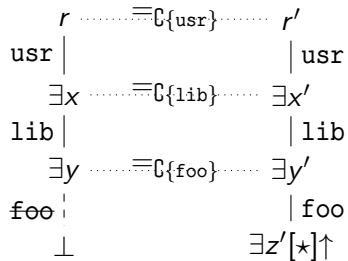
$r =_\star r'$
usr
$\exists x$
<del>lib</del>
$\perp$

$r =_\star r'$
usr
$\exists x$
lib
$\exists y$
foo
$\exists z$

Error cases

# Example Specification

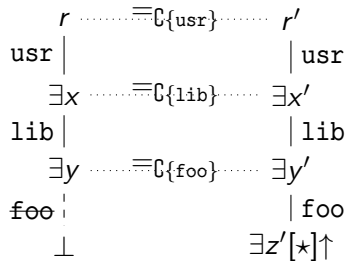
```
mkdir /usr/lib/foo
```



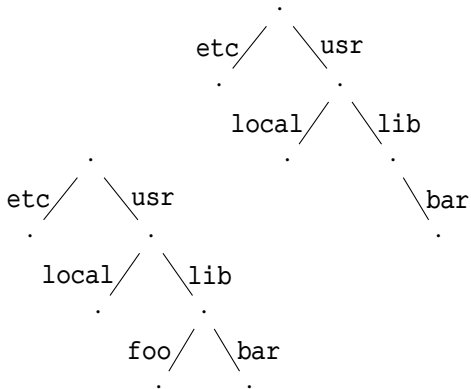
Success case

# Example Specification

mkdir /usr/lib/foo

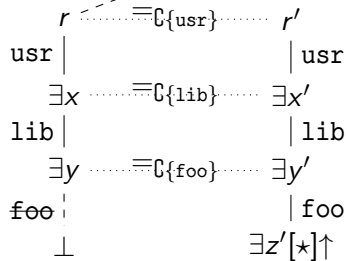


Success case

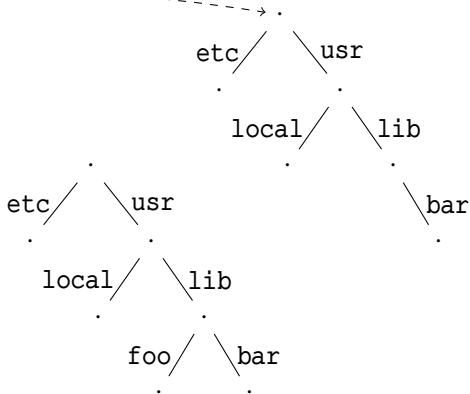


# Example Specification

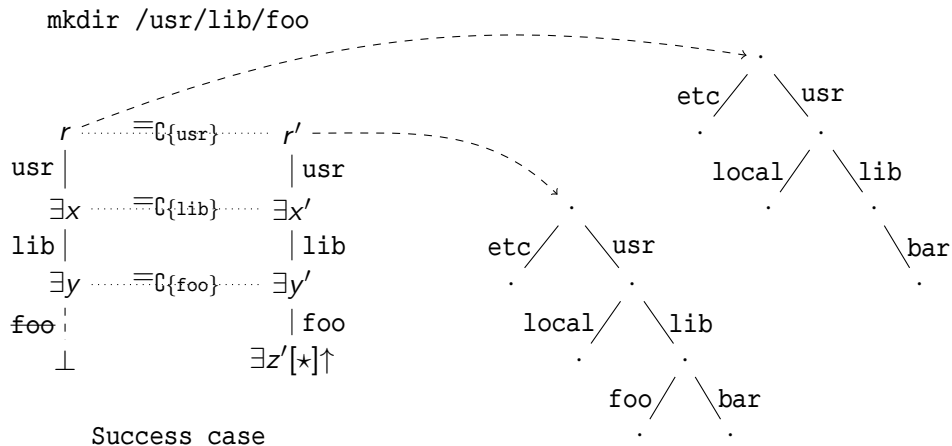
mkdir /usr/lib/foo



Success case



# Example Specification



## Transforming Formulas of FTS

# System of Transformation Rules – $\mathcal{R}_1$

*name*                      *pattern*     $\Rightarrow$     *replacement*    (*side-condition*)

System of Transformation Rules –  $\mathcal{R}_1$ 

*name*                      *pattern*     $\Rightarrow$     *replacement*    (*side-condition*)

	$x \dots =_F \dots y$		$x \dots =_F \dots y$	
P-Feat-Sim	$f \mid$	$\Rightarrow$	$f \backslash \quad / f$	$(f \in F)$
	$z$		$z$	



System of Transformation Rules –  $\mathcal{R}_1$ 

*name*                      *pattern*     $\Rightarrow$     *replacement*    (*side-condition*)

P-Feat-Sim                      
$$\begin{array}{c} x \dots =_F \dots y \\ f \mid \\ z \end{array} \Rightarrow \begin{array}{c} x \dots =_F \dots y \\ f \backslash \quad / f \\ z \end{array} \quad (f \in F)$$

D-Feats                      
$$\begin{array}{c} x \\ f / \quad \backslash f \\ y \quad \quad z \end{array} \Rightarrow \begin{array}{c} x \\ \mid f \\ y =_{\star} z \end{array}$$

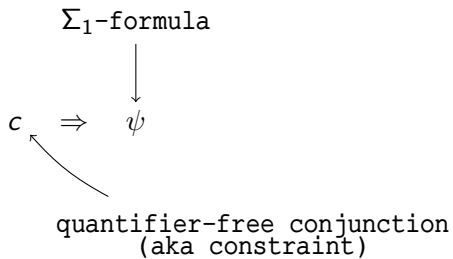
System of Transformation Rules –  $\mathcal{R}_1$ 

<i>name</i>	<i>pattern</i>	$\Rightarrow$	<i>replacement</i>	<i>(side-condition)</i>
P-Feat-Sim	$\begin{array}{c} x \dots =_F \dots y \\ f \mid \\ z \end{array}$	$\Rightarrow$	$\begin{array}{c} x \dots =_F \dots y \\ f \backslash \quad / f \\ z \end{array}$	$(f \in F)$
D-Feats	$\begin{array}{c} x \\ f / \quad \backslash f \\ y \quad \quad z \end{array}$	$\Rightarrow$	$\begin{array}{c} x \\ \mid f \\ y =_{\star} z \end{array}$	
C-Feat-Abs	$\begin{array}{c} x[F]\uparrow \\ f \mid \\ y \end{array}$	$\Rightarrow$	$\perp$	$(f \in F)$

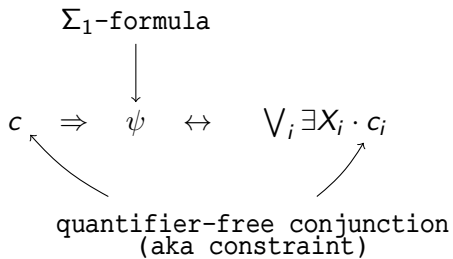
Transforming  $\Sigma_1$ -formulas

$$c \Rightarrow \psi$$

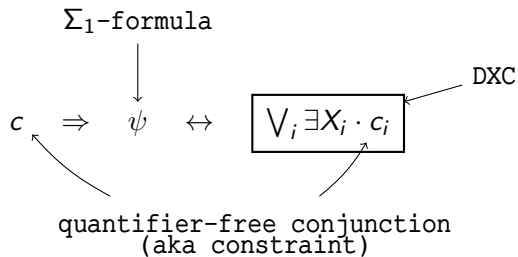
# Transforming $\Sigma_1$ -formulas



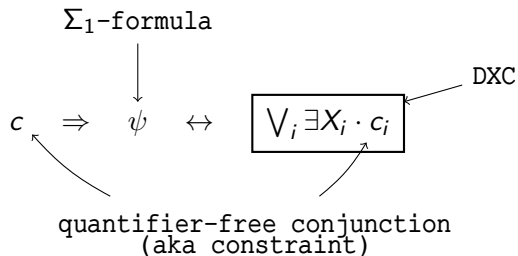
# Transforming $\Sigma_1$ -formulas



# Transforming $\Sigma_1$ -formulas



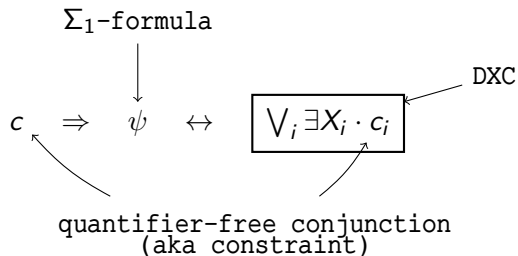
# Transforming $\Sigma_1$ -formulas



The rules of  $\mathcal{R}_1$  can be used to write:

```
function transform-1 ( $\phi$  :  $\Sigma_1$ -formula) : DXC
```

# Transforming $\Sigma_1$ -formulas



The rules of  $\mathcal{R}_1$  can be used to write:

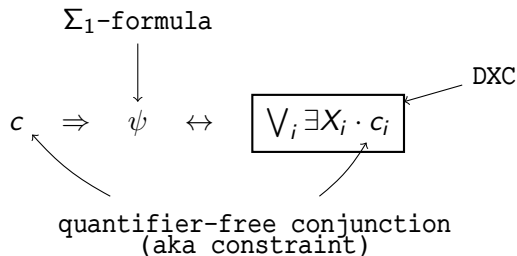
```
function transform-1 ( $\phi$  :  $\Sigma_1$ -formula) : DXC
```

which:

> terminates on all inputs\*,



# Transforming $\Sigma_1$ -formulas



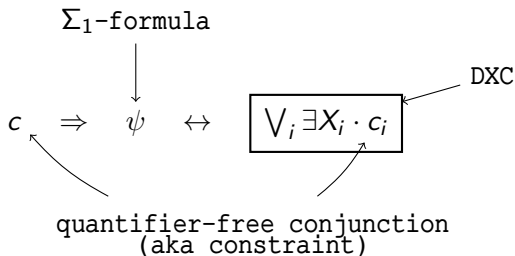
The rules of  $\mathcal{R}_1$  can be used to write:

```
function transform-1 ( $\phi$  :  $\Sigma_1$ -formula) : DXC
```

which:

- > terminates on all inputs\*,
- > given a  $\Sigma_1$ -formula  $\phi$ , returns a DXC  $\psi$  such that:
  - >  $\psi$  is equivalent to  $\phi$ ,

# Transforming $\Sigma_1$ -formulas



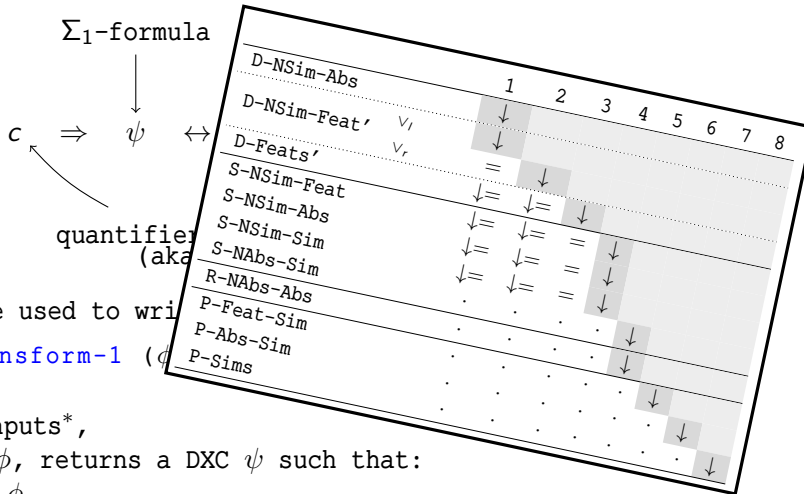
The rules of  $\mathcal{R}_1$  can be used to write:

```
function transform-1 ( $\phi$  :  $\Sigma_1$ -formula) : DXC
```

which:

- > terminates on all inputs\*,
- > given a  $\Sigma_1$ -formula  $\phi$ , returns a DXC  $\psi$  such that:
  - >  $\psi$  is equivalent to  $\phi$ ,
  - > and the constraints of  $\psi$  are irreducible with respect to  $\mathcal{R}_1$ .

# Transforming $\Sigma_1$ -formulas



The rules of  $\mathcal{R}_1$  can be used to write

`function transform-1 ( $\phi$ )`

which:

- > terminates on all inputs\*,
- > given a  $\Sigma_1$ -formula  $\phi$ , returns a DXC  $\psi$  such that:
  - >  $\psi$  is equivalent to  $\phi$ ,
  - > and the constraints of  $\psi$  are irreducible with respect to  $\mathcal{R}_1$ .

# What About Irreducible Constraints?

## What About Irreducible Constraints?

### Theorem (Satisfiability)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Then  $c$  is satisfiable.*

## What About Irreducible Constraints?

### Theorem (Satisfiability)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Then  $c$  is satisfiable.*

### Theorem (Garbage Collection)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .*

## What About Irreducible Constraints?

### Theorem (Satisfiability)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Then  $c$  is satisfiable.*

### Theorem (Garbage Collection)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Let  $X$  be a set of variables ancestor-closed in  $c$ .*

# What About Irreducible Constraints?

## Theorem (Satisfiability)


*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Then  $c$  is satisfiable.*

## Theorem (Garbage Collection)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Let  $X$  be a set of variables ancestor-closed in  $c$ .  
Then:*

$$\models \tilde{V} \cdot ((\exists X \cdot c) \leftrightarrow \mathcal{G}_X(c))$$

literals of  $c$   
that do not contain  
variables in  $X$





# What About Irreducible Constraints?

## Theorem (Satisfiability)

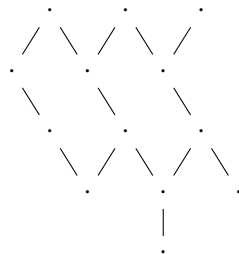
*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Then  $c$  is satisfiable.*

## Theorem (Garbage Collection)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Let  $X$  be a set of variables ancestor-closed in  $c$ .  
Then:*

$$\models \tilde{V} \cdot ((\exists X \cdot c) \leftrightarrow \mathcal{G}_X(c))$$

literals of  $c$   
that do not contain  
variables in  $X$



# What About Irreducible Constraints?

## Theorem (Satisfiability)

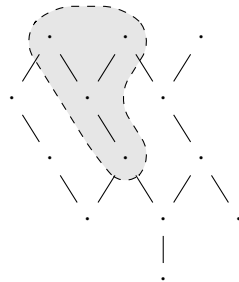
*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Then  $c$  is satisfiable.*

## Theorem (Garbage Collection)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Let  $X$  be a set of variables ancestor-closed in  $c$ .  
Then:*

$$\models \tilde{V} \cdot ((\exists X \cdot c) \leftrightarrow \mathcal{G}_X(c))$$

literals of  $c$   
that do not contain  
variables in  $X$



# What About Irreducible Constraints?

## Theorem (Satisfiability)

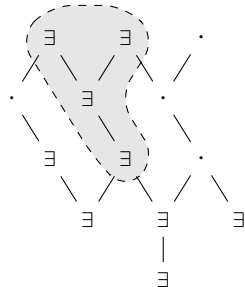
*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Then  $c$  is satisfiable.*

## Theorem (Garbage Collection)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Let  $X$  be a set of variables ancestor-closed in  $c$ .  
Then:*

$$\models \tilde{V} \cdot ((\exists X \cdot c) \leftrightarrow \mathcal{G}_X(c))$$

literals of  $c$   
that do not contain  
variables in  $X$



# What About Irreducible Constraints?

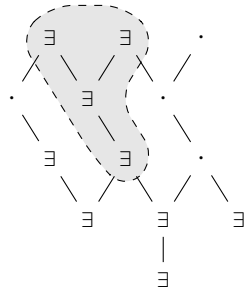
## Theorem (Satisfiability)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Then  $c$  is satisfiable.*

## Theorem (Garbage Collection)

*Let  $c$  be a constraint irreducible with respect to  $\mathcal{R}_1$ .  
Let  $X$  be a set of variables ancestor-closed in  $c$ .  
Then:*

$$\models \tilde{V} \cdot ((\exists X \cdot c) \leftrightarrow \mathcal{G}_X(c))$$



`transform-1` can be used as an incremental test of satisfiability!

# Composing Specifications

```
mkdir /usr/lib  
(success)
```

```
mkdir /usr/lib/foo  
(success)
```

## Composing Specifications

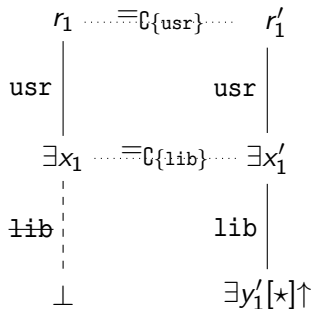
mkdir /usr/lib  
(success)

mkdir /usr/lib/foo  
(success)

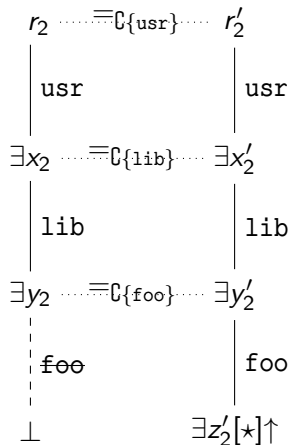
$$\begin{array}{ccc}
 r_1 & \dots \equiv_{\mathcal{C}\{\text{usr}\}} \dots & r'_1 \\
 \text{usr} \mid & & \text{usr} \mid \\
 \exists x_1 & \dots \equiv_{\mathcal{C}\{\text{lib}\}} \dots & \exists x'_1 \\
 \text{lib} \mid & & \text{lib} \mid \\
 \vdots & & \vdots \\
 \perp & & \exists y'_1[\star]\uparrow
 \end{array}$$

## Composing Specifications

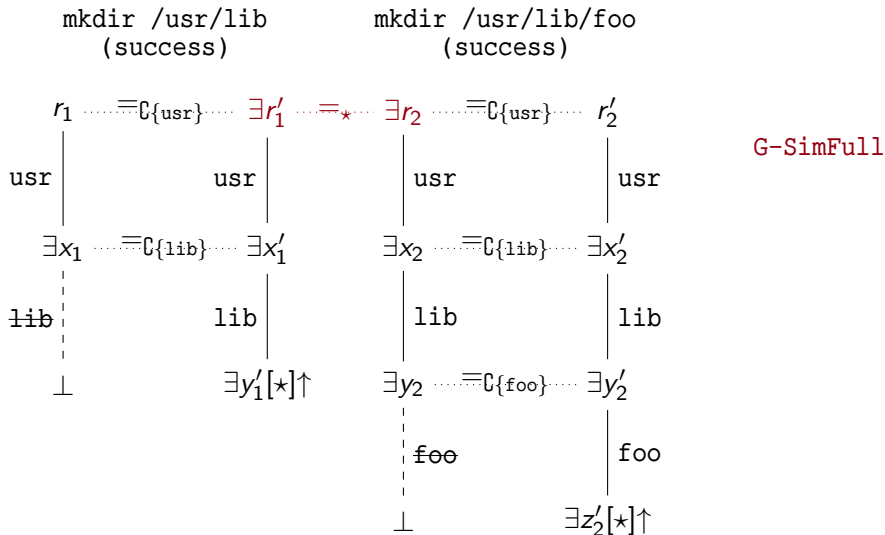
mkdir /usr/lib  
(success)



mkdir /usr/lib/foo  
(success)

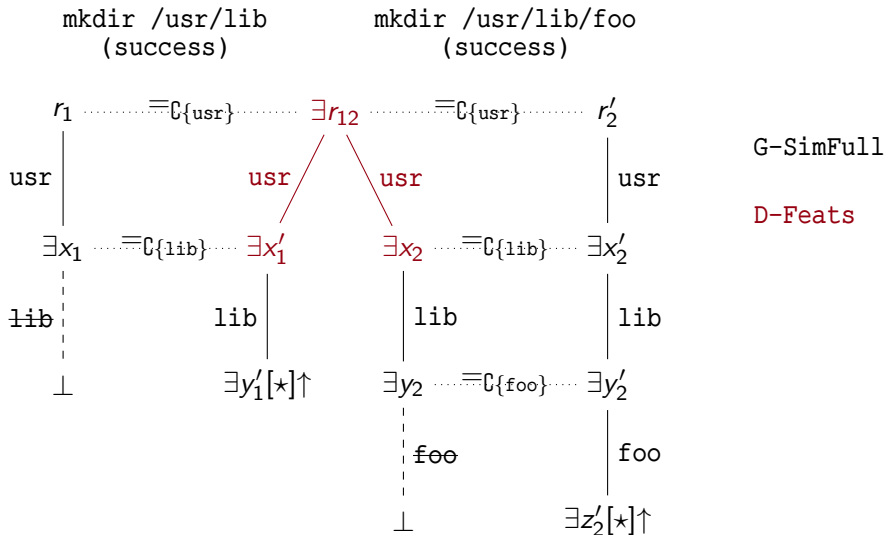


## Composing Specifications

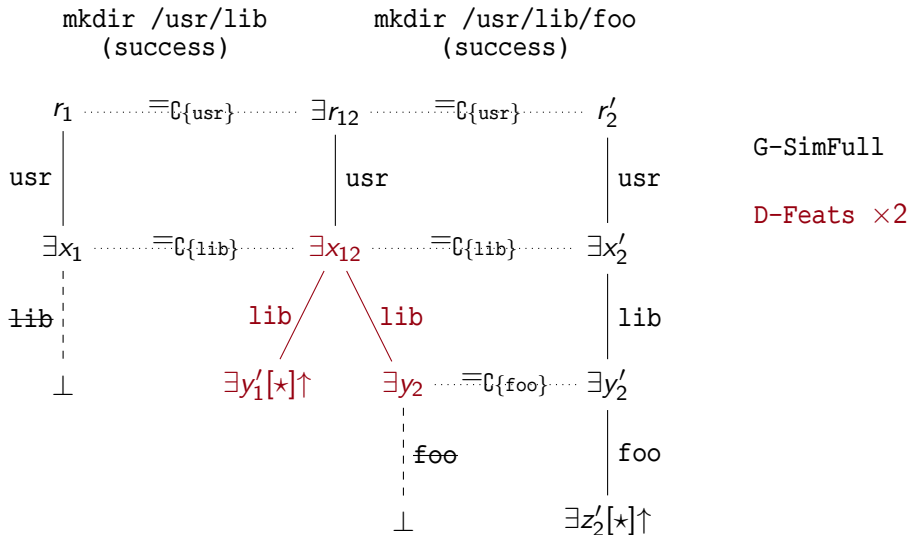




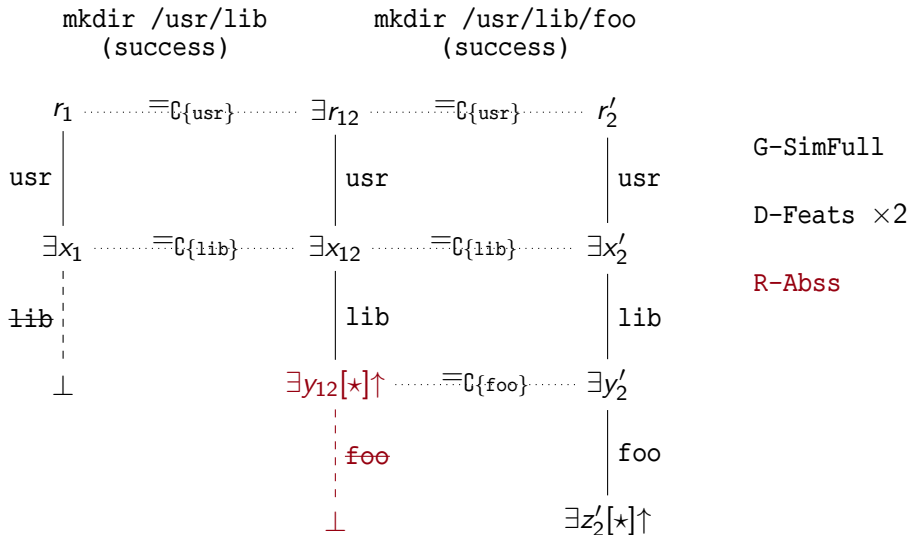
## Composing Specifications



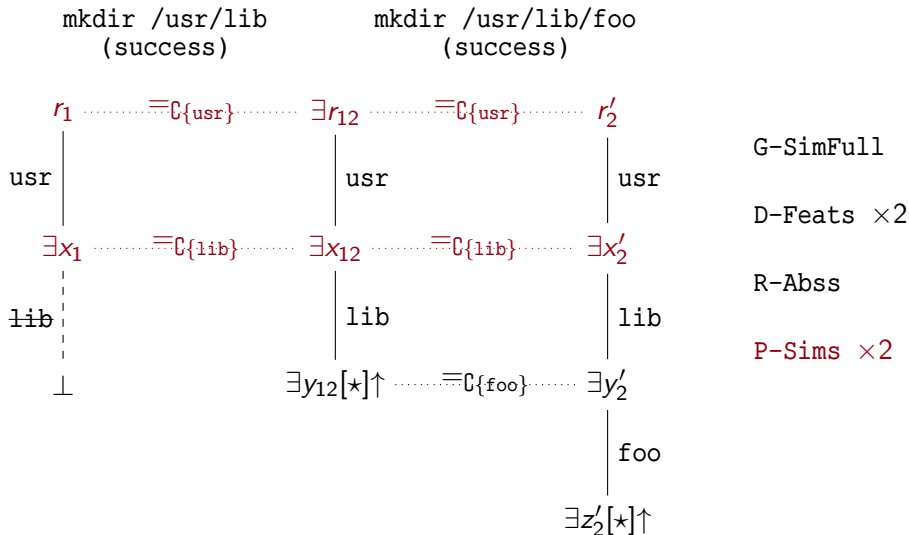
## Composing Specifications



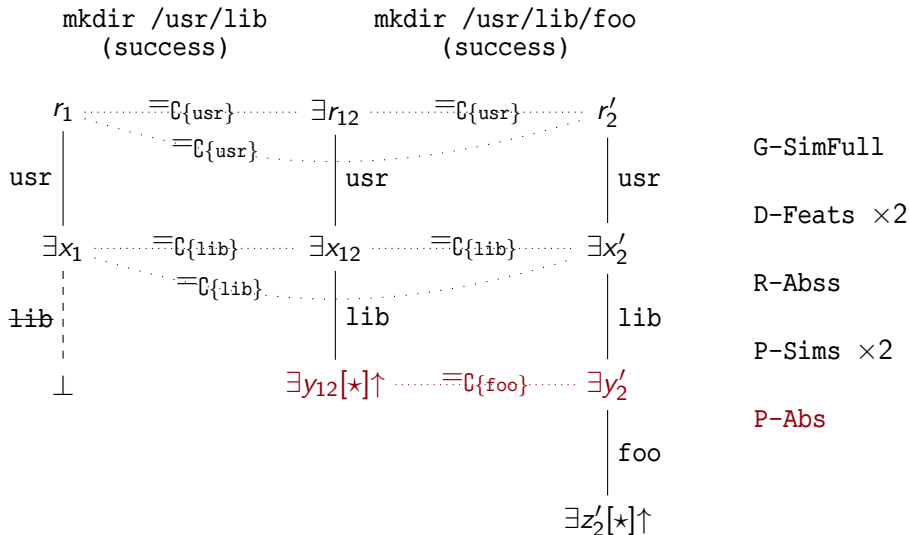
## Composing Specifications



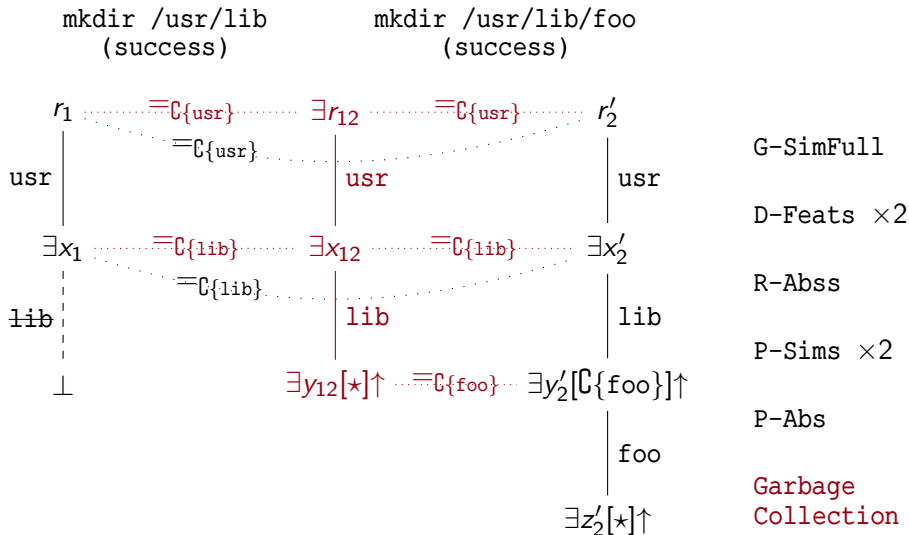
## Composing Specifications



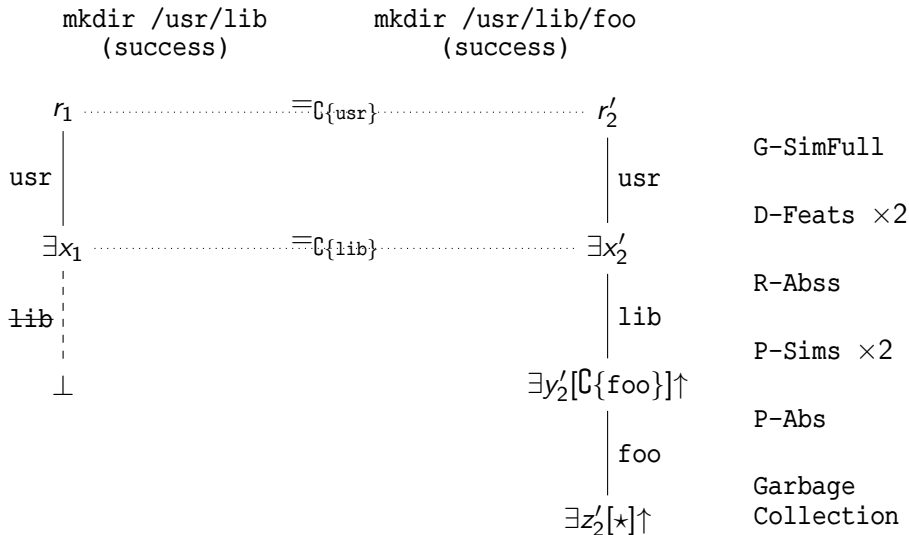
## Composing Specifications



## Composing Specifications



## Composing Specifications



# Composing Specifications

```
mkdir /usr/lib  
(success)
```

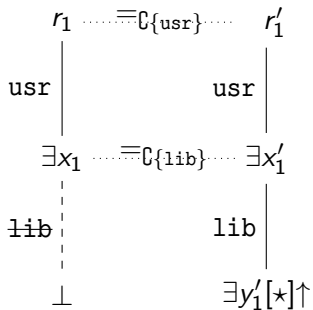
```
mkdir /usr/lib/foo  
(error: file exists)
```



## Composing Specifications

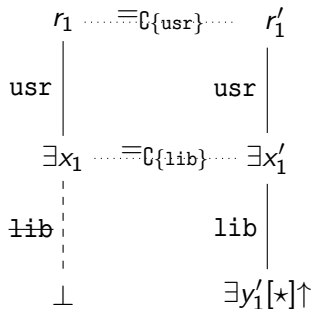
mkdir /usr/lib  
(success)

mkdir /usr/lib/foo  
(error: file exists)

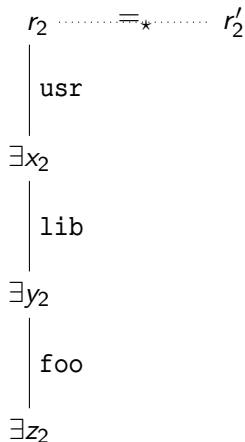


## Composing Specifications

mkdir /usr/lib  
(success)



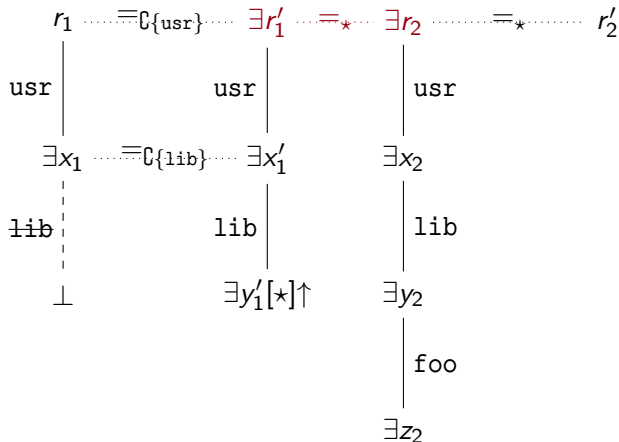
mkdir /usr/lib/foo  
(error: file exists)



## Composing Specifications

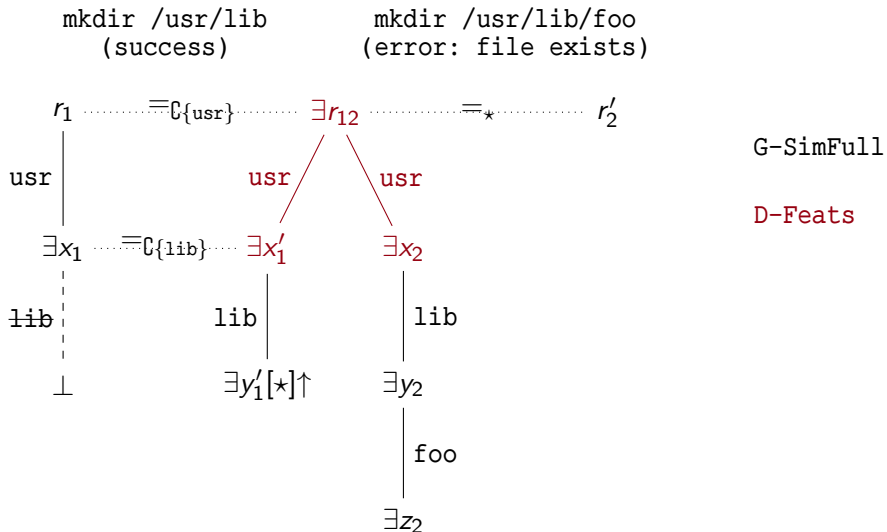
mkdir /usr/lib  
(success)

mkdir /usr/lib/foo  
(error: file exists)



G-SimFull

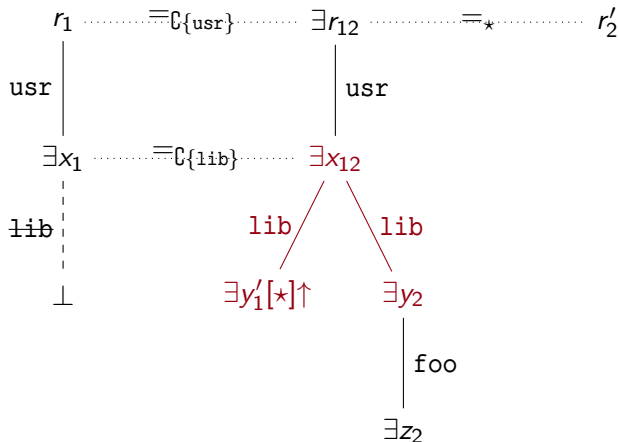
## Composing Specifications



## Composing Specifications

mkdir /usr/lib  
(success)

mkdir /usr/lib/foo  
(error: file exists)



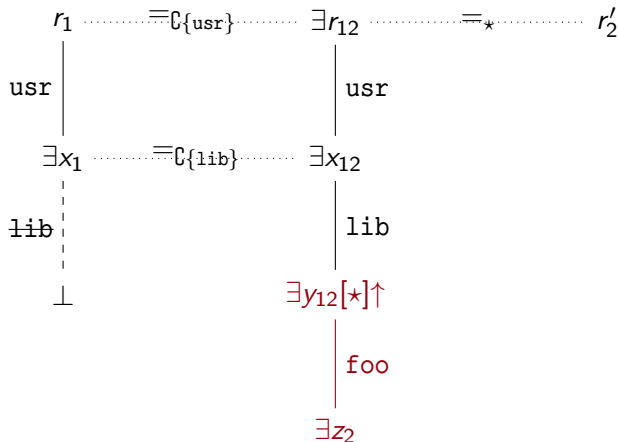
G-SimFull

D-Feats  $\times 2$

## Composing Specifications

mkdir /usr/lib  
(success)

mkdir /usr/lib/foo  
(error: file exists)



G-SimFull

D-Feats  $\times 2$

C-Feat-Abs

# Decidability of the First-order Theory of FTS

# Decidability of the First-order Theory of FTS

> no quantifier elimination in the strict sense

$$\exists y \cdot (x[f]y \wedge y[\star]\uparrow)$$



# Decidability of the First-order Theory of FTS

- > no quantifier elimination in the strict sense

$$\exists y \cdot (x[f]y \wedge y[\star]\uparrow)$$

- > but: garbage collection removes existential quantifiers

# Decidability of the First-order Theory of FTS

- > no quantifier elimination in the strict sense

$$\exists y \cdot (x[f]y \wedge y[\star]\uparrow)$$

- > but: garbage collection removes existential quantifiers
- > the others can be switched to universal

$$\neg x[f]\uparrow \wedge \forall y \cdot (x[f]y \rightarrow y[\star]\uparrow)$$

# Decidability of the First-order Theory of FTS

- > no quantifier elimination in the strict sense

$$\exists y \cdot (x[f]y \wedge y[\star]\uparrow)$$

- > but: garbage collection removes existential quantifiers
- > the others can be switched to universal

$$\neg x[f]\uparrow \wedge \forall y \cdot (x[f]y \rightarrow y[\star]\uparrow)$$

- > in general, we can transform any  $\Sigma_1$ -formula into a  $\Pi_1$  one

# Decidability of the First-order Theory of FTS

- > no quantifier elimination in the strict sense

$$\exists y \cdot (x[f]y \wedge y[\star]\uparrow)$$

- > but: garbage collection removes existential quantifiers
- > the others can be switched to universal

$$\neg x[f]\uparrow \wedge \forall y \cdot (x[f]y \rightarrow y[\star]\uparrow)$$

- > in general, we can transform any  $\Sigma_1$ -formula into a  $\Pi_1$  one
- > we can apply a “weak quantifier elimination”

# Decidability of the First-order Theory of FTS

- > no quantifier elimination in the strict sense

$$\exists y. (x[f]y \wedge y[*]\uparrow)$$

- > but: garbage collection
- > the others can be switched

## Theorem

*The first-order theory of FTS is decidable.*

$$\neg x[f]\uparrow \wedge \forall y. (x[f]y \rightarrow y[f]\uparrow)$$

- > in general, we can transform any  $\Sigma_1$ -formula into a  $\Pi_1$  one
- > we can apply a "weak quantifier elimination"

## Efficient Solving of Constraints in FTS

## And in Practice?

> `transform-1` is not really usable in practice!

## And in Practice?

- > `transform-1` is not really usable in practice!
- > symbolic execution: sensitive to combinatorial explosion



## And in Practice?

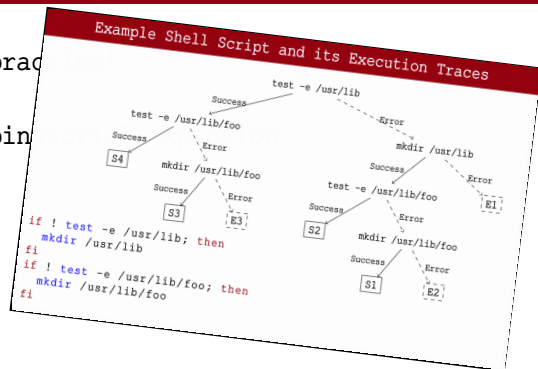
- > `transform-1` is not really usable in practice!
- > symbolic execution: sensitive to combinatorial explosion
- > three sources of such explosions:

## And in Practice?

- > `transform-1` is not really usable in practice!
- > symbolic execution: sensitive to combinatorial explosion
- > three sources of such explosions:
  - > traces of execution

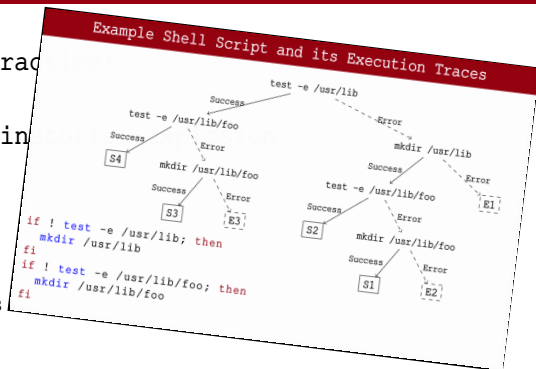
# And in Practice?

- > `transform-1` is not really usable in practice
- > symbolic execution: sensitive to combination of
- > three sources of such explosions:
  - > traces of execution
  - > recall the tree of traces



## And in Practice?

- > transform-1 is not really usable in practice
- > symbolic execution: sensitive to combination of options
- > three sources of such explosions:
  - > traces of execution
    - > recall the tree of traces
    - > mitigated by pruning unreachable traces



## And in Practice?

- > `transform-1` is not really usable in practice!
- > symbolic execution: sensitive to combinatorial explosion
- > three sources of such explosions:
  - > traces of execution
    - > recall the tree of traces
    - > mitigated by pruning unreachable traces
  - > rules of  $\mathcal{R}_1$

# And in Practice?

- > `transform-1` is not really usable in practice!
- > symbolic execution: sensitive to combinatorial explosion
- > three sources of such explosions:
  - > traces of execution
    - > recall the tree of traces
    - > mitigated by pruning unreachable traces
  - > rules of  $\mathcal{R}_1$

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow$$

# And in Practice?

- > `transform-1` is not really usable in practice!
- > symbolic execution: sensitive to combinatorial explosion
- > three sources of such explosions:
  - > traces of execution
    - > recall the tree of traces
    - > mitigated by pruning unreachable traces
  - > rules of  $\mathcal{R}_1$

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$

# And in Practice?

- > `transform-1` is not really usable in practice!
- > symbolic execution: sensitive to combinatorial explosion
- > three sources of such explosions:
  - > traces of execution
    - > recall the tree of traces
    - > mitigated by pruning unreachable traces
  - > rules of  $\mathcal{R}_1$

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$

- > specifications



## And in Practice?

> `transform-1` is not really usable in practice!

> symbolic execution: sensitive to

> three sources of such explosions:

> traces of execution

> recall the tree of traces

> mitigated by pruning unreachable

> rules of  $\mathcal{R}_1$

D-NFeat

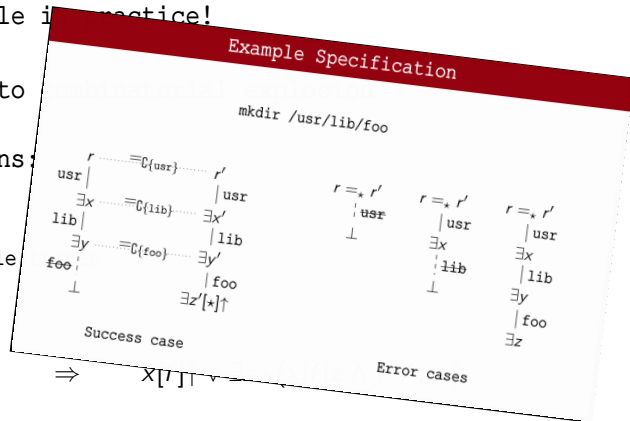
$\neg x[f]y$

$\Rightarrow$

$X[T] \vdash$

> specifications

> recall the specification of "`mkdir /usr/lib/foo`" with three error cases



A New System of Rules –  $\mathcal{R}_2$ 

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$

---

## A New System of Rules – $\mathcal{R}_2$

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$

---

> new system of rules  $\mathcal{R}_2$

## A New System of Rules – $\mathcal{R}_2$

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$

---

- > new system of rules  $\mathcal{R}_2$
- > holds on to such literals until we know which side to choose

A New System of Rules –  $\mathcal{R}_2$ 

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$


---

- > new system of rules  $\mathcal{R}_2$
- > holds on to such literals until we know which side to choose

$$\text{D-NFeat-Abs} \quad \neg x[f]y \wedge x[f]\uparrow \quad \Rightarrow \quad x[f]\uparrow$$

# A New System of Rules – $\mathcal{R}_2$

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$


---

- > new system of rules  $\mathcal{R}_2$
- > holds on to such literals until we know which side to choose

$$\begin{array}{lll} \text{D-NFeat-Abs} & \neg x[f]y \wedge x[f]\uparrow & \Rightarrow \quad x[f]\uparrow \\ \text{D-NFeat-Feat} & \neg x[f]y \wedge x[f]z & \Rightarrow \quad x[f]z \wedge y \neq_\star z \end{array}$$

# A New System of Rules – $\mathcal{R}_2$

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$


---

- > new system of rules  $\mathcal{R}_2$
- > holds on to such literals until we know which side to choose

$$\begin{array}{llll} \text{D-NFeat-Abs} & \neg x[f]y \wedge x[f]\uparrow & \Rightarrow & x[f]\uparrow \\ \text{D-NFeat-Feat} & \neg x[f]y \wedge x[f]z & \Rightarrow & x[f]z \wedge y \neq_\star z \end{array}$$

- >  $\mathcal{R}_2$  never introduces disjunctions or variables

# A New System of Rules – $\mathcal{R}_2$

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$


---

- > new system of rules  $\mathcal{R}_2$
- > holds on to such literals until we know which side to choose

$$\begin{array}{llll} \text{D-NFeat-Abs} & \neg x[f]y \wedge x[f]\uparrow & \Rightarrow & x[f]\uparrow \\ \text{D-NFeat-Feat} & \neg x[f]y \wedge x[f]z & \Rightarrow & x[f]z \wedge y \neq_\star z \end{array}$$

- >  $\mathcal{R}_2$  never introduces disjunctions or variables
- > we loose the garbage collection of irreducible constraints



# A New System of Rules – $\mathcal{R}_2$

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$


---

- > new system of rules  $\mathcal{R}_2$
- > holds on to such literals until we know which side to choose

$$\begin{array}{llll} \text{D-NFeat-Abs} & \neg x[f]y \wedge x[f]\uparrow & \Rightarrow & x[f]\uparrow \\ \text{D-NFeat-Feat} & \neg x[f]y \wedge x[f]z & \Rightarrow & x[f]z \wedge y \neq_\star z \end{array}$$

- >  $\mathcal{R}_2$  never introduces disjunctions or variables
- > we loose the garbage collection of irreducible constraints
  - > and  $\mathcal{R}_2$  is therefore useless to decide the first-order

# A New System of Rules – $\mathcal{R}_2$

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$


---

- > new system of rules  $\mathcal{R}_2$
- > holds on to such literals until we know which side to choose

$$\begin{array}{lll} \text{D-NFeat-Abs} & \neg x[f]y \wedge x[f]\uparrow & \Rightarrow \quad x[f]\uparrow \\ \text{D-NFeat-Feat} & \neg x[f]y \wedge x[f]z & \Rightarrow \quad x[f]z \wedge y \neq_\star z \end{array}$$

- >  $\mathcal{R}_2$  never introduces disjunctions or variables
- > we loose the garbage collection of irreducible constraints
  - > and  $\mathcal{R}_2$  is therefore useless to decide the first-order
  - > but we can recover garbage collection partially

# A New System of Rules – $\mathcal{R}_2$

**Problem 1:**  $\mathcal{R}_1$  introduces disjunctions and new variables.

$$\text{D-NFeat} \quad \neg x[f]y \quad \Rightarrow \quad x[f]\uparrow \vee \exists z \cdot (x[f]z \wedge y \neq_\star z)$$


---

- > new system of rules  $\mathcal{R}_2$
- > holds on to such literals until we know which side to choose

$$\begin{array}{llll} \text{D-NFeat-Abs} & \neg x[f]y \wedge x[f]\uparrow & \Rightarrow & x[f]\uparrow \\ \text{D-NFeat-Feat} & \neg x[f]y \wedge x[f]z & \Rightarrow & x[f]z \wedge y \neq_\star z \end{array}$$

- >  $\mathcal{R}_2$  never introduces disjunctions or variables
- > we loose the garbage collection of irreducible constraints
  - > and  $\mathcal{R}_2$  is therefore useless to decide the first-order
  - > but we can recover garbage collection partially
- > we can still prove satisfiability of irreducible constraints

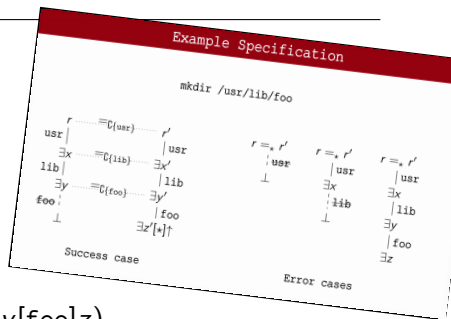
# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---

## Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.



$$\begin{aligned} & r[\text{usr}] \uparrow \\ & \vee \exists x. (r[\text{usr}]x \wedge x[\text{lib}] \uparrow) \\ & \vee \exists x, y, z. (r[\text{usr}]x \wedge x[\text{lib}]y \wedge y[\text{foo}]z) \end{aligned}$$

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---

Threaded  
Constraint

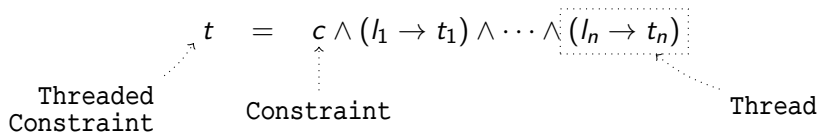
$$t = c \wedge (l_1 \rightarrow t_1) \wedge \cdots \wedge (l_n \rightarrow t_n)$$

$$\begin{aligned} & r[\text{usr}] \uparrow \\ & \vee \exists x \cdot (r[\text{usr}]x \wedge x[\text{lib}] \uparrow) \\ & \vee \exists x, y, z \cdot (r[\text{usr}]x \wedge x[\text{lib}]y \wedge y[\text{foo}]z) \end{aligned}$$

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---

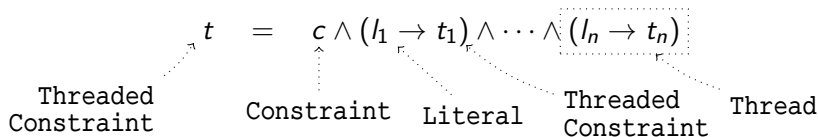


$$\begin{aligned}
 & r[\text{usr}] \uparrow \\
 & \vee \exists x \cdot (r[\text{usr}]x \wedge x[\text{lib}] \uparrow) \\
 & \vee \exists x, y, z \cdot (r[\text{usr}]x \wedge x[\text{lib}]y \wedge y[\text{foo}]z)
 \end{aligned}$$

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---



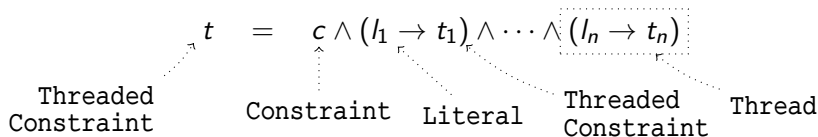
$$\begin{aligned}
 & r[\text{usr}] \uparrow \\
 & \vee \exists x \cdot (r[\text{usr}]x \wedge x[\text{lib}] \uparrow) \\
 & \vee \exists x, y, z \cdot (r[\text{usr}]x \wedge x[\text{lib}]y \wedge y[\text{foo}]z)
 \end{aligned}$$



# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---



$$\begin{aligned}
 & r[\text{usr}] \uparrow \\
 & \vee \exists x \cdot (r[\text{usr}]x \wedge x[\text{lib}] \uparrow) \\
 & \vee \exists x, y, z \cdot (r[\text{usr}]x \wedge x[\text{lib}]y \wedge y[\text{foo}]z)
 \end{aligned}$$

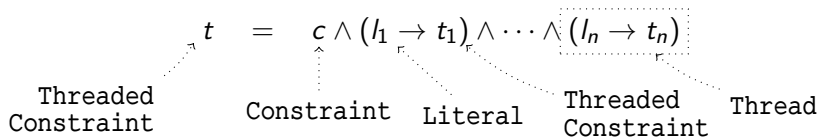
becomes:

$$\neg r[\text{usr}] \uparrow \rightarrow$$

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---



$$\begin{aligned}
 & r[\text{usr}] \uparrow \\
 & \vee \exists x \cdot (r[\text{usr}]x \wedge x[\text{lib}] \uparrow) \\
 & \vee \exists x, y, z \cdot (r[\text{usr}]x \wedge x[\text{lib}]y \wedge y[\text{foo}]z)
 \end{aligned}$$

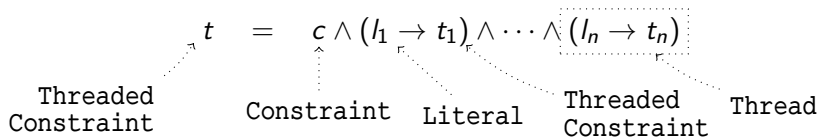
becomes:

$$\neg r[\text{usr}] \uparrow \rightarrow \exists x \cdot (r[\text{usr}]x \quad )$$

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---



$$\begin{aligned}
 & r[\text{usr}] \uparrow \\
 & \vee \exists x \cdot (r[\text{usr}]x \wedge x[\text{lib}] \uparrow) \\
 & \vee \exists x, y, z \cdot (r[\text{usr}]x \wedge x[\text{lib}]y \wedge y[\text{foo}]z)
 \end{aligned}$$

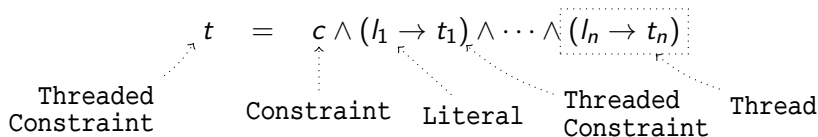
becomes:

$$\neg r[\text{usr}] \uparrow \rightarrow \exists x \cdot (r[\text{usr}]x \wedge (\neg x[\text{lib}] \uparrow \rightarrow \quad ))$$

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---



$$\begin{aligned}
 & r[\text{usr}] \uparrow \\
 & \vee \exists x \cdot (r[\text{usr}]x \wedge x[\text{lib}] \uparrow) \\
 & \vee \exists x, y, z \cdot (r[\text{usr}]x \wedge x[\text{lib}]y \wedge y[\text{foo}]z)
 \end{aligned}$$

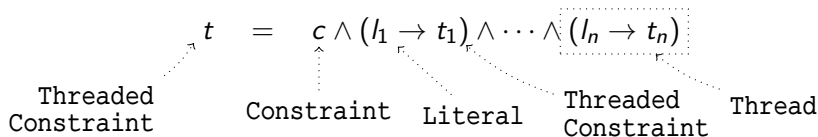
becomes:

$$\neg r[\text{usr}] \uparrow \rightarrow \exists x \cdot (r[\text{usr}]x \wedge (\neg x[\text{lib}] \uparrow \rightarrow \exists y \cdot (x[\text{lib}]y \quad )))$$

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---



$$\begin{aligned}
 & r[\text{usr}] \uparrow \\
 & \vee \exists x \cdot (r[\text{usr}]x \wedge x[\text{lib}] \uparrow) \\
 & \vee \exists x, y, z \cdot (r[\text{usr}]x \wedge x[\text{lib}]y \wedge y[\text{foo}]z)
 \end{aligned}$$

becomes:

$$\neg r[\text{usr}] \uparrow \rightarrow \exists x \cdot (r[\text{usr}]x \wedge (\neg x[\text{lib}] \uparrow \rightarrow \exists y \cdot (x[\text{lib}]y \wedge \exists z \cdot y[\text{foo}]z)))$$

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---

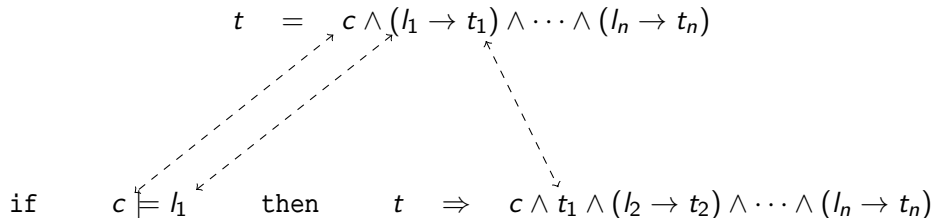
$$t = c \wedge (l_1 \rightarrow t_1) \wedge \cdots \wedge (l_n \rightarrow t_n)$$

if  $c \models l_1$

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

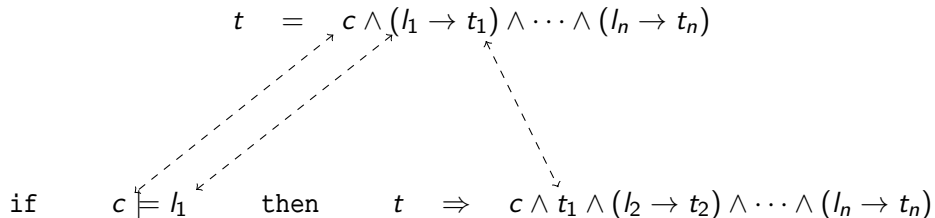
---



# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---



Run on toolchain:

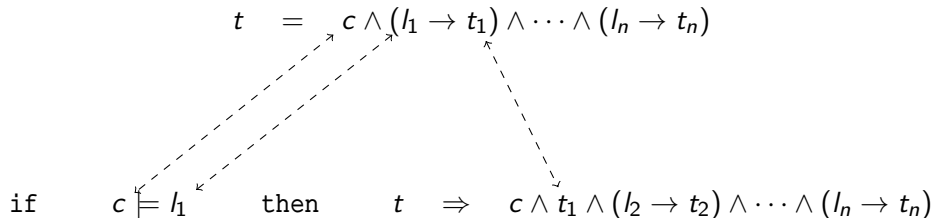
- > on 113,328 scenarios
- > with a timeout of 60s



# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---



Run on toolchain:

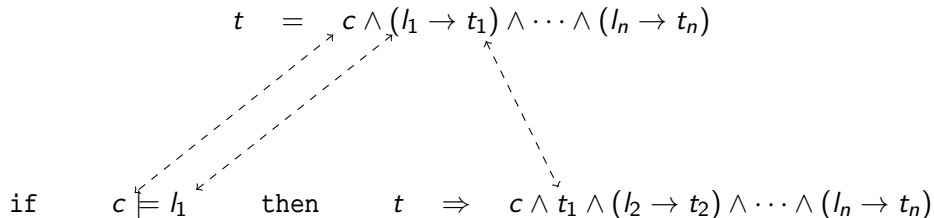
- > on 113,328 scenarios
- > with a timeout of 60s

	Without	With
Time	8h25	
Scenarios	45%	

# Threaded Constraints

**Problem 2:** Constraints are not expressive enough for specifications.

---



Run on toolchain:

- > on 113,328 scenarios
- > with a timeout of 60s

	Without	With
Time	8h25	0h22
Scenarios	45%	52%

## Applications

<http://the.report/rancid-cgi/>

## Report > rancid-cgi

### Meta

Start time  
2021-01-29 08:56:30  
End time  
2021-01-29 08:56:30  
Duration  
0s

### Parsing Status

Name  
rancid-cgi  
Version  
3.10-1  
Maintainer scripts

preinst

OK

postinst

Rejected by conversion unsupported feature: special builtin: exec

<http://the.report/rancid-cgi/>

## Report > rancid-cgi

### Meta

Start time  
2021-01-29 08:56:30  
End time  
2021-01-29 08:56:30  
Duration  
0s

### Parsing Status

Name  
rancid-cgi  
Version  
3.10-1  
Maintainer scripts

**preinst**

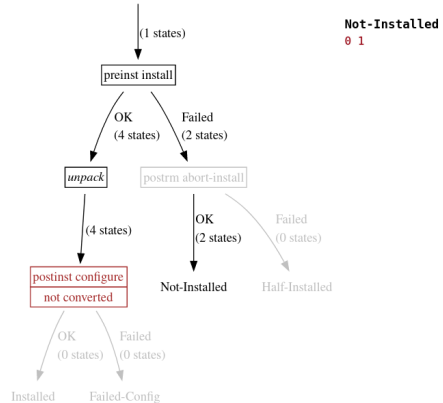
OK

**postinst**

Rejected by conversion unsupported feature: special builtin: exec

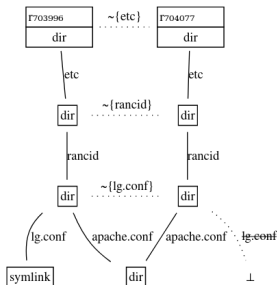
## Scenarios Summaries

### Installation



<http://the.report/rancid-cgi/install/not-installed/1.html>

## Report > rancid-cgi > Installation > Not-Installed #1



### log

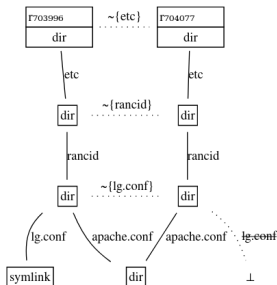
```

[TRACE] test -h /etc/rancid/lg.conf: path resolves to file of type 'l'
[TRACE] rm /etc/rancid/lg.conf: remove file
[TRACE] test -e /etc/rancid/apache.conf: path resolves
[TRACE] rm /etc/rancid/apache.conf: target does not exist or is a di

```

<http://the.report/rancid-cgi/install/not-installed/1.html>

## Report > rancid-cgi > Installation > Not-Installed #1



### log

```
[TRACE] test -h /etc/rancid/lg.conf: path resolves to file of type 'l'
[TRACE] rm /etc/rancid/lg.conf: remove file
[TRACE] test -e /etc/rancid/apache.conf: path resolves
[TRACE] rm /etc/rancid/apache.conf: target does not exist or is a di
```

### Original Shell script

```
1 #! /bin/sh
2 # preinst script for rancid
3 #
4 # see: dh_installdeb(1)
5
6 set -e
7
8 # To remove old bad env link
9 if [ -h /etc/rancid/lg.conf ]; then
10     rm /etc/rancid/lg.conf
11fi
12if [ -e /etc/rancid/apache.conf ]; then
13     rm /etc/rancid/apache.conf
14fi
15
16
17# dh_installdeb will replace this with shell code automatically
18# generated by other debhelper scripts.
19
20
21
22exit 0
```

# Bugs Found

Bugs	Closed	Detected by	Examples
95	56	parser	not using -e mode
6	4	parser & manual	unsafe or non-POSIX constructs
34	24	corpus mining	wrong options, mixed redirections
9	7	conversion	wrong test expressions
5	2	symbolic execution	try to remove a directory with rm
3	3	formalisation	bug in dpkg-maintscript-helper
151	92		



## Conclusion & Future Work

# Conclusion

> Theoretical results:

# Conclusion

- > Theoretical results:
  - > FTS

# Conclusion

- > Theoretical results:
  - > FTS
  - > Decision procedures

# Conclusion

- > Theoretical results:
  - > FTS
  - > Decision procedures
  - > Decidability of the first-order

# Conclusion

- > Theoretical results:
  - > FTS
  - > Decision procedures
  - > Decidability of the first-order
- > Modelisation:

# Conclusion

- > Theoretical results:
  - > FTS
  - > Decision procedures
  - > Decidability of the first-order
- > Modelisation:
  - > POSIX Shell

# Conclusion

- > Theoretical results:
  - > FTS
  - > Decision procedures
  - > Decidability of the first-order
  
- > Modelisation:
  - > POSIX Shell
  - > Unix filesystems & utilities



# Conclusion

- > Theoretical results:
  - > FTS
  - > Decision procedures
  - > Decidability of the first-order
  
- > Modelisation:
  - > POSIX Shell
  - > Unix filesystems & utilities
  
- > Implementation:

# Conclusion

- > Theoretical results:
  - > FTS
  - > Decision procedures
  - > Decidability of the first-order
  
- > Modelisation:
  - > POSIX Shell
  - > Unix filesystems & utilities
  
- > Implementation:
  - > Parser & Conversion for POSIX Shell

# Conclusion

- > Theoretical results:
  - > FTS
  - > Decision procedures
  - > Decidability of the first-order
- > Modelisation:
  - > POSIX Shell
  - > Unix filesystems & utilities
- > Implementation:
  - > Parser & Conversion for POSIX Shell
  - > Efficient solver for FTS

# Conclusion

- > Theoretical results:
  - > FTS
  - > Decision procedures
  - > Decidability of the first-order
  
- > Modelisation:
  - > POSIX Shell
  - > Unix filesystems & utilities
  
- > Implementation:
  - > Parser & Conversion for POSIX Shell
  - > Efficient solver for FTS
  - > Toolchain scaling to 30,000 packages

## Future Work on our Toolchain

- > Support for more maintainer scripts:

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications
- > Automated testing of specifications:



## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications
- > Automated testing of specifications:
  - > take a utility call `c`

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications
- > Automated testing of specifications:
  - > take a utility call  $c$
  - > take its specification  $\phi(r, r')$

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications
  
- > Automated testing of specifications:
  - > take a utility call  $c$
  - > take its specification  $\phi(r, r')$
  - > take  $fs$  such that  $[r \mapsto fs] \models \exists r' \cdot \phi(r, r')$

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications
- > Automated testing of specifications:
  - > take a utility call  $c$
  - > take its specification  $\phi(r, r')$
  - > take  $fs$  such that  $[r \mapsto fs] \models \exists r' \cdot \phi(r, r')$
  - > run  $c$  on  $fs$ , obtain  $fs'$

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications
  
- > Automated testing of specifications:
  - > take a utility call  $c$
  - > take its specification  $\phi(r, r')$
  - > take  $fs$  such that  $[r \mapsto fs] \models \exists r' \cdot \phi(r, r')$
  - > run  $c$  on  $fs$ , obtain  $fs'$
  - > check that  $[r \mapsto fs, r' \mapsto fs'] \models \phi(r, r')$

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications
  
- > Automated testing of specifications:
  - > take a utility call  $c$
  - > take its specification  $\phi(r, r')$
  - > take  $fs$  such that  $[r \mapsto fs] \models \exists r' \cdot \phi(r, r')$
  - > run  $c$  on  $fs$ , obtain  $fs'$
  - > check that  $[r \mapsto fs, r' \mapsto fs'] \models \phi(r, r')$
  
- > Generalisation:

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications
  
- > Automated testing of specifications:
  - > take a utility call  $c$
  - > take its specification  $\phi(r, r')$
  - > take  $fs$  such that  $[r \mapsto fs] \models \exists r' \cdot \phi(r, r')$
  - > run  $c$  on  $fs$ , obtain  $fs'$
  - > check that  $[r \mapsto fs, r' \mapsto fs'] \models \phi(r, r')$
  
- > Generalisation:
  - > to other distributions

## Future Work on our Toolchain

- > Support for more maintainer scripts:
  - > extend intermediary language
  - > write more specifications
- > Automated testing of specifications:
  - > take a utility call  $c$
  - > take its specification  $\phi(r, r')$
  - > take  $fs$  such that  $[r \mapsto fs] \models \exists r' \cdot \phi(r, r')$
  - > run  $c$  on  $fs$ , obtain  $fs'$
  - > check that  $[r \mapsto fs, r' \mapsto fs'] \models \phi(r, r')$
- > Generalisation:
  - > to other distributions
  - > to Shell scripts in general



## Future Work on FTS

- > Extension of FTS to support:

## Future Work on FTS

- > Extension of FTS to support:
  - > quantification over features;

## Future Work on FTS

- > Extension of FTS to support:
  - > quantification over features;
  - > paths of features;

## Future Work on FTS

- > Extension of FTS to support:
  - > quantification over features;
  - > paths of features;
  - > inclusion of trees ( $x \subseteq y$ ).

## Future Work on FTS

- > Extension of FTS to support:
  - > quantification over features;
  - > paths of features;
  - > inclusion of trees ( $x \subseteq y$ ).
- > A solver that supports globs:

## Future Work on FTS

- > Extension of FTS to support:
  - > quantification over features;
  - > paths of features;
  - > inclusion of trees ( $x \subseteq y$ ).
- > A solver that supports globs:
  - > to specify commands like: `rm *.foo`

# Future Work on FTS

- > Extension of FTS to support:
  - > quantification over features;
  - > paths of features;
  - > inclusion of trees ( $x \subseteq y$ ).
  
- > A solver that supports globs:
  - > to specify commands like: `rm *.foo`
  - > with:  $\neg r[*.foo] \uparrow \wedge r'[*.foo] \uparrow \wedge r = \mathbb{C}_{\{*.foo\}} r'$ .

# Future Work on FTS

- > Extension of FTS to support:
  - > quantification over features;
  - > paths of features;
  - > inclusion of trees ( $x \subseteq y$ ).
- > A solver that supports globs:
  - > to specify commands like: `rm *.foo`
  - > with:  $\neg r[*.foo] \uparrow \wedge r'[*.foo] \uparrow \wedge r = \mathbb{C}_{\{*.foo\}} r'$ .
- > "Efficient" decision procedures for:



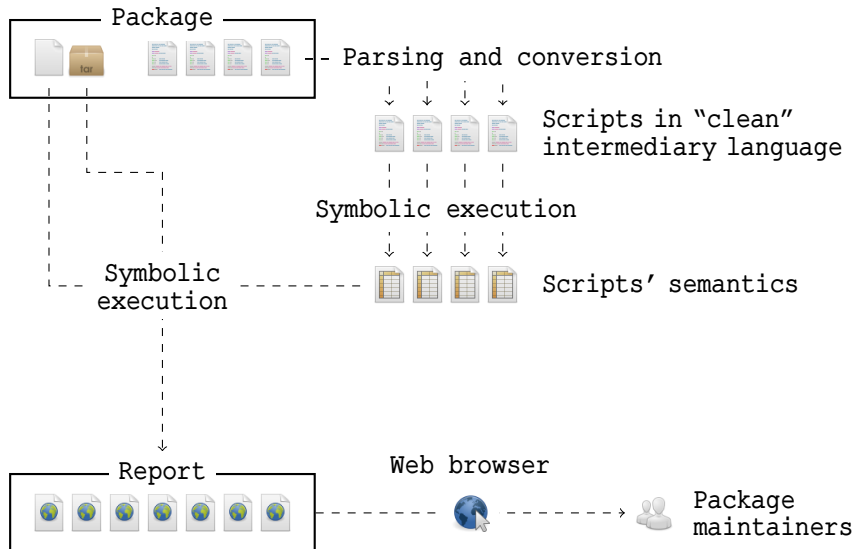
# Future Work on FTS

- > Extension of FTS to support:
  - > quantification over features;
  - > paths of features;
  - > inclusion of trees ( $x \subseteq y$ ).
- > A solver that supports globs:
  - > to specify commands like: `rm *.foo`
  - > with:  $\neg r[*.foo] \uparrow \wedge r'[*.foo] \uparrow \wedge r = \mathbb{C}_{\{*.foo\}} r'$ .
- > "Efficient" decision procedures for:
  - > first-order,

# Future Work on FTS

- > Extension of FTS to support:
  - > quantification over features;
  - > paths of features;
  - > inclusion of trees ( $x \subseteq y$ ).
- > A solver that supports globs:
  - > to specify commands like: `rm *.foo`
  - > with:  $\neg r[*.foo] \uparrow \wedge r'[*.foo] \uparrow \wedge r = \mathbb{C}_{\{*.foo\}} r'$ .
- > "Efficient" decision procedures for:
  - > first-order,
  - > entailment of  $\Sigma_1$ -formulas.

# Thank You!



## Contributions:

- > parser for POSIX Shell\*
- > intermediary language\*
- > feature trees logic FTS
- > decision procedures
- > decidability results
- > efficiency considerations
- > implementation