APT 3.0 dependency solver

An orthodox approach to dependency solving, leading to a SAT solver comparable to DPLL.

Julian Andres Klode

Canonical Ltd

Contents

Introduction

Definitions

Facts

Let

- $\mathcal{V}$ be the set of versions in the apt cache (literals)
- $\mathcal{I} \subset \mathcal{V}$ be the set of installed versions
- $\mathcal{M} \subset \mathcal{I}$ be the set of manually installed versions
- $\mathcal{A} = \mathcal{I} \setminus \mathcal{M}$ be the set of automatically installed versions

Let

- $\mathcal{D}_V \subset \{D1 \vee \ldots \vee D_n \mid D_1, \ldots, D_n \in \mathcal{V}\}$
- $\mathcal{C}_V \subset \{C \mid C \in \mathcal{V}\}$

denote the dependencies and conflicts of $V \in \mathcal{V}$. These correspond to the formulas: $V \to D1 \vee \ldots \vee D_n$ and $V \to \neg C$.

(TODO: Optional dependencies)

Let $|D_1 \vee \ldots \vee D_n| = n$ represent the number of choices in a given "or group".

Solver state

For depth $i \in \mathbb{N}$, and step $j \in \mathbb{N}$:

Let

- $needs_{ij} \subset \mathcal{V}$ denote the set of versions that shall be installed
- $rejects_{ij} \subset \mathcal{V}$ denote the set of versions that shall not be installed
- $wants_ij \subset \mathcal{V}$ denote the set of versions that we want installed later (optional dependencies)
- $likes_{ij} \subset \mathcal{V}$ denote the set of versions that are also suggested by packages (more optional)

Let $allversions(V)$ denote the ordered set of all (allowed for install) versions of the package that V is a version of.

Let $work_{ij} \subset \{V \to D \mid D \in \mathcal{D}V\}$ denote the work queue of unsatisfied dependencies.

Let $needs_{00} = rejects_{00} = wants_{00} = likes_{00} = \emptyset$.

Iteration

Let the symbol $\perp$ determine termination of the solver (mostly fatal), and $\top$ denote termination of that level.

$$needs_{i,j+1} = \begin{cases} \perp & \text{if } \forall d \in w : d \in rejects_{ij} \\ needs_{ij} & \text{if } \exists d \in w : d \in needs_{ij} \text{ (already installed)} \\ needs_{ij} \cup d & \text{if } \exists d \in w : d = \{w\} \text{ (single choice)} \\ \top & \text{otherwise} \end{cases}$$

$$rejects_{i,j+1} = \begin{cases} \perp & \text{if } needs_{i,j+1} = \perp \\ \top & \text{if } needs_{i,j+1} = \top \\ rejects_{ij} \cup \{d \in \mathcal{C}_d\} & \text{otherwise} \end{cases}$$