

Пожидаев Михаил Сергеевич

Deepsolver — инструмент управления пакетами
с программным обеспечением

1. Доставка с удалённых узлов, установка, обновление и удаление пакетов.
2. Наполнение автоматизированной среды для компиляции исходных текстов.
3. Поиск доступных пакетов в удалённых репозиториях и предоставление информации о них.
4. Поддержание целостности операционной системы (ОС).

1. *OperationCore* — обработка запросов пользователя на внесение изменений в ОС.
2. *InfoCore* — поиск информации о доступных пакетах.
3. *IndexCore* — построение индексов репозитория.

Две альтернативы при работе с NP-полными задачами:

- ▶ применить приближённый алгоритм без гарантии нахождения решения;
- ▶ наложить дополнительные ограничения и применить точный полиномиальный алгоритм.

Уравнение для приближённого алгоритма

$$(!p \vee r_1 \vee r_2 \vee r_3) \wedge (!p \vee r_4 \vee r_5) \wedge (!p \vee r_6) \wedge (!p \vee !c_1) \wedge (!p \vee !c_2),$$

где p — рассматриваемый пакет,

r_1, r_2, r_3 — варианты разрешения некоторой зависимости R_1 ,

r_4, r_5 — варианты разрешения некоторой зависимости R_2 ,

r_6 — пакет для разрешения некоторой зависимости R_3 ,

а c_1 и c_2 — пакеты, конфликтующие с p .

Уравнение для полиномиального алгоритма

$$(!p \vee r_1) \wedge (!p \vee r_4) \wedge (!p \vee r_6) \wedge (!p \vee !c_1) \wedge (!p \vee !c_2),$$

где p — рассматриваемый пакет,

r_1 — выбранное разрешение некоторой зависимости R_1 ,

r_4 — выбранное разрешение некоторой зависимости R_2 ,

r_6 — пакет для разрешения некоторой зависимости R_3 ,

а c_1 и c_2 — пакеты, конфликтующие с p .

$(!p \vee r_1) \wedge (!p \vee r_4) \wedge (!p \vee r_6) \wedge (!p \vee !c_1) \wedge (!p \vee !c_2)$ — решается полиномиально.

$(!p \vee r_1)$ заменяется парой импликаций $p \rightarrow r_1$ и $!r_1 \rightarrow !p$.

Граф импликаций

Каждая переменная заменяется парой вершин для положительного и отрицательного значения, каждая импликация заменяется дугой.

Если существует путь из положительного значения в отрицательное, переменной необходимо присвоить отрицательное значение. Если существует обратный путь или пути нет вообще, то выбирается положительное значение.

Если существуют оба пути — уравнение решения не имеет! Проверка выполняется алгоритмом поиска компонент сильной связности.

Откуда берётся p ?

1. При нарушении зависимости на пакет в случае его обновления произвести обновление зависимых пакетов.
2. Произвести замену пакета, используя правила обработки зависимостей, если результат вычислений не будет совпадать с удаляемым или обновляемым пакетом.
3. Полностью удалить зависимое поддерево установленных пакетов.

Спасибо за внимание!