Верификация моделей программ

ЛЕКТОР:

Владимир Анатольевич Захаров
Владислав Васильевич Подымов

zakh@cs.msu.su

Лекция 11.

Ограниченная верификация моделей программ (Bounded model checking)

1. **Разнообразие методов верификации моделей программ**
2. **Задача проверки выполнимости булевых формул (SAT) и способы ее решения**
3. **Задача ограниченной верификации моделей программ (BMC)**
4. **Трансляция задачи BMC в задачу SAT**

Разнообразие методов верификации моделей программ

Неоспоримый тезис

В основе каждого способа решения задачи верификации моделей программ лежит некоторая задача дискретной математики и методы ее решения.

Разнообразие методов верификации моделей программ

Неоспоримый тезис

В основе каждого способа решения задачи верификации моделей программ лежит некоторая задача дискретной математики и методы ее решения.

1. Табличный и теоретико-автоматный методы . Модели представляются в виде графа. Верификация модели сводится к проверке достижимости заданного множества вершин и компонент связности в графе.

Разнообразие методов верификации моделей программ

Неоспоримый тезис

В основе каждого способа решения задачи верификации моделей программ лежит некоторая задача дискретной математики и методы ее решения.

1. Табличный и теоретико-автоматный методы . Модели представляются в виде графа. Верификация модели сводится к проверке достижимости заданного множества вершин и компонент связности в графе.
2. Символьный метод . Модели представляются в виде ROBDD. Верификация модели сводится к вычислению неподвижных точек операторов на конечных множествах посредством операций над ROBDD.

Разнообразие методов верификации моделей программ

1. Табличный и теоретико-автоматный методы .
   Преимущества. Устроен просто и работает быстро.
   Трудности. Применим только к моделям небольшого размера ($10^8 - 10^{10}$ состояний).

Разнообразие методов верификации моделей программ

1. Табличный и теоретико-автоматный методы .
   Преимущества. Устроен просто и работает быстро.
   Трудности. Применим только к моделям небольшого размера ($10^8 - 10^{10}$ состояний).
2. Символьный метод .
   Преимущества. Применим к моделям большого размера (10^{100} и более состояний).
   Трудности. Требует тонкой настройки — выбора подходящей абстракции при построении модели, оптимального упорядочения переменных при построении ROBDD, большого объема памяти для ее хранения.

Разнообразие методов верификации моделей программ

1. Табличный и теоретико-автоматный методы .
   Преимущества. Устроен просто и работает быстро.
   Трудности. Применим только к моделям небольшого размера ($10^8 - 10^{10}$ состояний).
2. Символьный метод .
   Преимущества. Применим к моделям большого размера (10^{100} и более состояний).
   Трудности. Требует тонкой настройки — выбора подходящей абстракции при построении модели, оптимального упорядочения переменных при построении ROBDD, большого объема памяти для ее хранения.

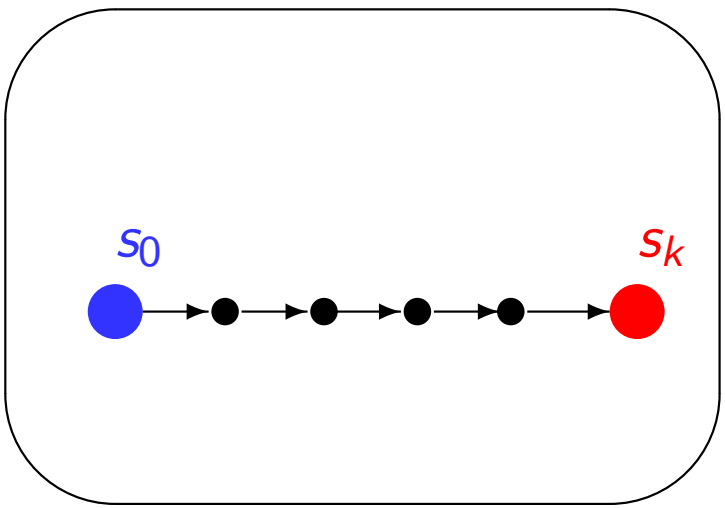
«С умным - хлопотно, с дураком - плохо.
Нужно что-то среднее.
Да где ж его взять?»

Разнообразие методов верификации моделей программ

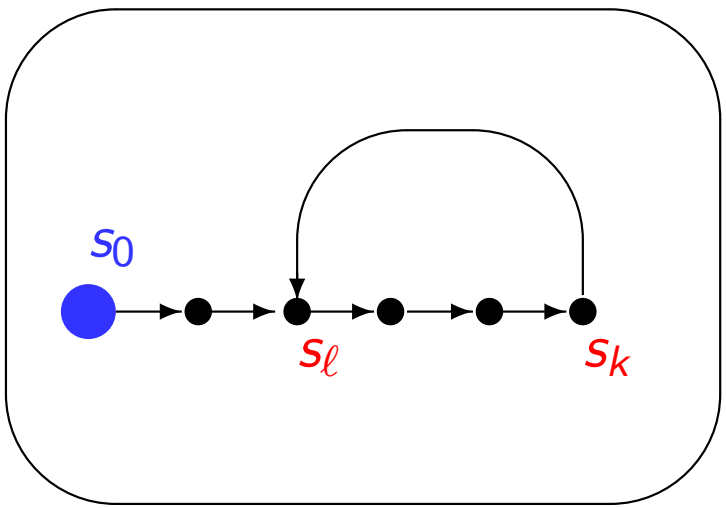
Идея

На практике чаще всего приходится проверять простые требования безопасности и живости, нарушение которых подтверждается обнаружением сравнительно короткого контр-примера — трассы длины $10 - 100$.

Safety counterexample



Liveness counterexample

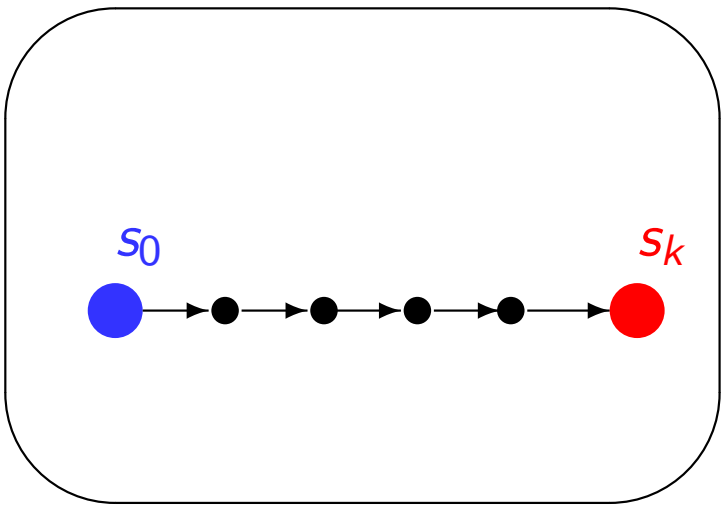


Разнообразие методов верификации моделей программ

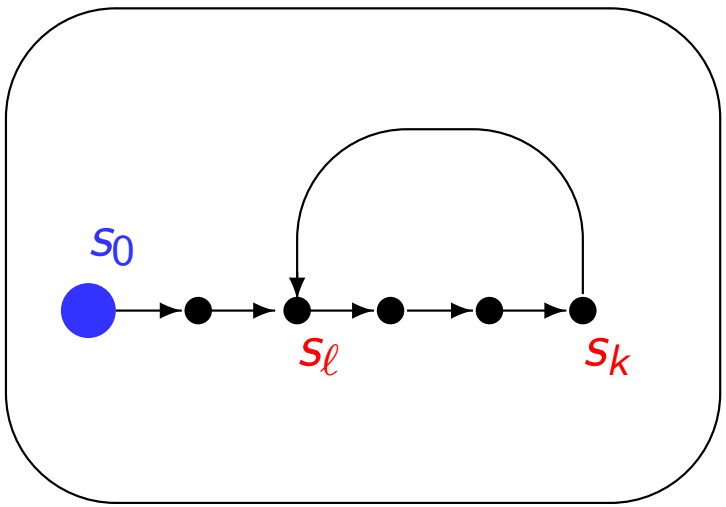
Идея

Предполагаемый контр-пример можно описать булевой формулой, утверждающей о том, что в модели существует последовательность переходов $s_0 \to s_1 \to \cdots \to s_k$ заданной длины k , проходящей через состояния определенного вида.

Safety counterexample



Liveness counterexample



Разнообразие методов верификации моделей программ

Идея

Построенная формула φ должна быть выполнима в том и только том случае, когда в модели есть «короткая» трасса, которая является контр-примером для проверяемого свойства.

Однако такая формула может зависеть от десятков тысяч переменных (число переменных состояния $\times$ длина контр-примера), и поэтому проверить ее выполнимость при помощи ROBDD практически невозможно — не хватит памяти.

Нам нужны специальные методы и средства решения задачи SAT проверки выполнимости булевых формул.

В чем же состоит эта задача и как она решается?

Задача проверки выполнимости булевых формул (SAT) и способы ее решения

Проблема выполнимости для булевых формул (SAT): для произвольной заданной булевой формулы $\varphi(x_1, x_2, \ldots, x_n)$ выяснить, существует ли такой набор значений переменных $x_1 = \sigma_1, x_2 = \sigma_2, \ldots, x_n = \sigma_n$, для которого верно равенство $\varphi(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1$

Задача проверки выполнимости булевых формул (SAT) и способы ее решения

Проблема выполнимости для булевых формул (SAT): для произвольной заданной булевой формулы $\varphi(x_1, x_2, \ldots, x_n)$ выяснить, существует ли такой набор значений переменных $x_1 = \sigma_1, x_2 = \sigma_2, \ldots, x_n = \sigma_n$, для которого верно равенство $\varphi(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1$

Замечание 1.

Можно считать, что в формуле $\varphi(x_1, x_2, \ldots, x_n)$ используются только операции $\neg, \wedge, \vee$.

Задача проверки выполнимости булевых формул (SAT) и способы ее решения

Проблема выполнимости для булевых формул (SAT): для произвольной заданной булевой формулы $\varphi(x_1, x_2, \ldots, x_n)$ выяснить, существует ли такой набор значений переменных $x_1 = \sigma_1, x_2 = \sigma_2, \ldots, x_n = \sigma_n$, для которого верно равенство $\varphi(\sigma_1, \sigma_2, \ldots, \sigma_n) = 1$

Замечание 1.

Можно считать, что в формуле $\varphi(x_1, x_2, \ldots, x_n)$ используются только операции $\neg, \wedge, \vee$.

Замечание 2.

Можно считать, что формула $\varphi(x_1, x_2, \ldots, x_n)$ — это конъюнктивная нормальная форма вида

$$3CNF = \bigwedge_{(i,j,k)\in I} (x_i^{\delta_i} \vee x_j^{\delta_j} \vee x_k^{\delta_k})$$

Задача SAT и способы ее решения

Иллюстрация к замечанию 2.

$$\Phi(\dots x_1 \to x_2 \dots)$$

Задача SAT и способы ее решения

Иллюстрация к замечанию 2.

$$\Phi(\dots x_1 \to x_2 \dots)$$

$$\Phi(\dots y \dots) \wedge (y \equiv (x_1 \to x_2))$$

Задача SAT и способы ее решения

Иллюстрация к замечанию 2.

$$\Phi(\dots x_1 \to x_2 \dots)$$

$$\Phi(\dots y \dots) \wedge (y \equiv (x_1 \to x_2))$$

$$\Phi(\dots y \dots) \wedge (y \vee x_1 \vee x_2) \wedge (y \vee x_1 \vee \overline{x_2}) \wedge (y \vee \overline{x_1} \vee \overline{x_2}) \wedge (\overline{y} \vee \overline{x_1} \vee x_2)$$

Задача SAT и способы ее решения

Иллюстрация к замечанию 2.

$$\Phi(\dots x_1 \to x_2 \dots)$$

$$\Phi(\dots y \dots) \wedge (y \equiv (x_1 \to x_2))$$

$$\Phi(\dots y \dots) \wedge (y \vee x_1 \vee x_2) \wedge (y \vee x_1 \vee \overline{x_2}) \wedge (y \vee \overline{x_1} \vee \overline{x_2}) \wedge (\overline{y} \vee \overline{x_1} \vee x_2)$$

Таким образом, любую булеву формулу с n переменными и m операциями можно преобразовать в равновыполнимую 3-КНФ с $n + m$ переменными и $4m$ дизъюнктами-сомножителями.

Задача SAT и способы ее решения

Как известно, проблема выполнимости 3-КНФ является NP-полной задачей.

Задача SAT и способы ее решения

Как известно, проблема выполнимости 3-КНФ является NP-полной задачей.

В свете современных математических знаний это означает, что

1. нет эффективной процедуры гарантированного решения задачи выполнимости 3-КНФ;

Задача SAT и способы ее решения

Как известно, проблема выполнимости 3-КНФ является NP-полной задачей.

В свете современных математических знаний это означает, что

1. нет эффективной процедуры гарантированного решения задачи выполнимости 3-КНФ;
2. сколь-нибудь эффективный метод решения задачи выполнимости 3-КНФ можно использовать для решения большого числа разнообразных задач комбинаторики, дискретной математики, алгебры, теории формальных языков и др.

Вот поэтому программы решения задачи выполнимости 3-КНФ (SAT-solvers) пользуются большим спросом.

Задача SAT и способы ее решения

Наиболее распространенные подходы к решению задачи SAT

- **метод DPLL с разрешением конфликтов**,
- **метод случайного блуждания**.

Задача SAT и способы ее решения

Наиболее распространенные подходы к решению задачи SAT

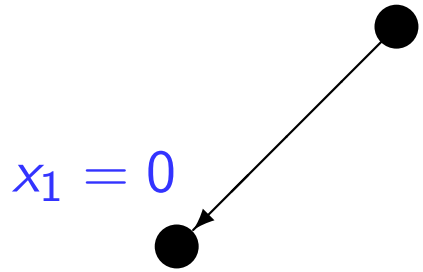
- **метод DPLL с разрешением конфликтов**,
- **метод случайного блуждания**.

DPLL (Davis–Putnam–Logemann–Loveland algorithm) — это переборный поиск набора значений переменных, на котором выполняется 3-КНФ, путем последовательного выбора подходящих значений для отдельных переменных и упрощения 3-КНФ в соответствии с выбранными значениями.

Задача SAT и способы ее решения

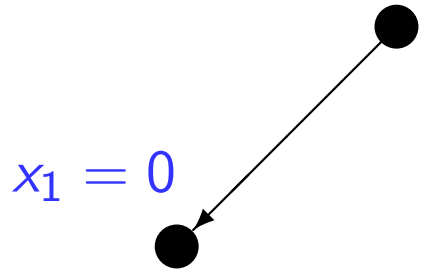
$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_3) \wedge (\overline{x_2} \vee x_4 \vee x_5) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5})$$

Задача SAT и способы ее решения



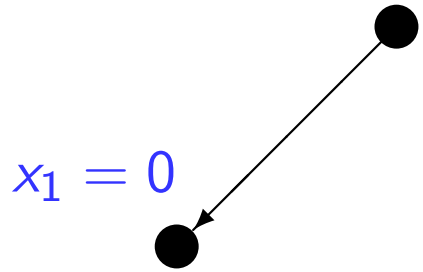
$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_3) \wedge (\overline{x_2} \vee x_4 \vee x_5) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5})$$

Задача SAT и способы ее решения



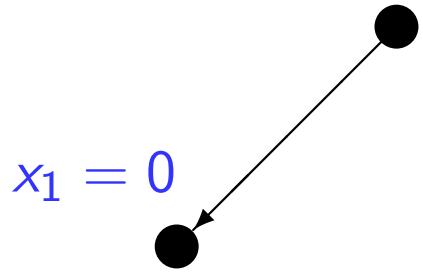
$$(0 \vee x_2) \wedge \ (\overline{0} \vee x_3) \wedge \ (\overline{x_2} \vee x_4 \vee x_5) \wedge (0 \vee \overline{x_2} \vee \overline{x_4}) \wedge \ (\overline{x_2} \vee \overline{x_5})$$

Задача SAT и способы ее решения



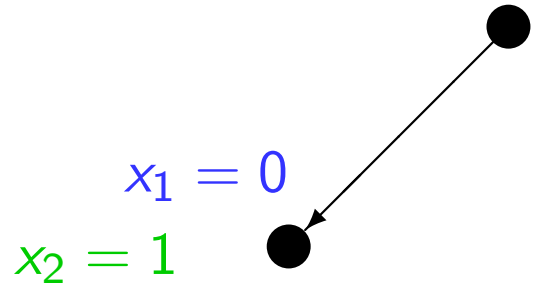
$x_2 \wedge \qquad (\overline{x_2} \vee x_4 \vee x_5) \wedge \quad (\overline{x_2} \vee \overline{x_4}) \wedge \quad (\overline{x_2} \vee \overline{x_5})$

Задача SAT и способы ее решения



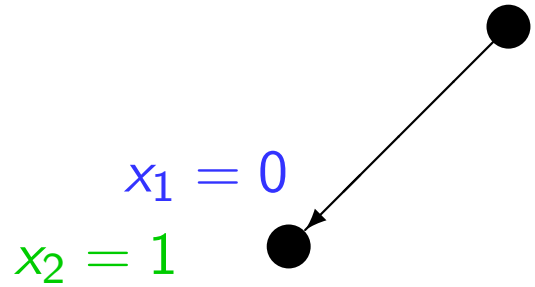
$x_2 \wedge \qquad (\overline{x_2} \vee x_4 \vee x_5) \wedge \quad (\overline{x_2} \vee \overline{x_4}) \wedge \quad (\overline{x_2} \vee \overline{x_5})$

Задача SAT и способы ее решения



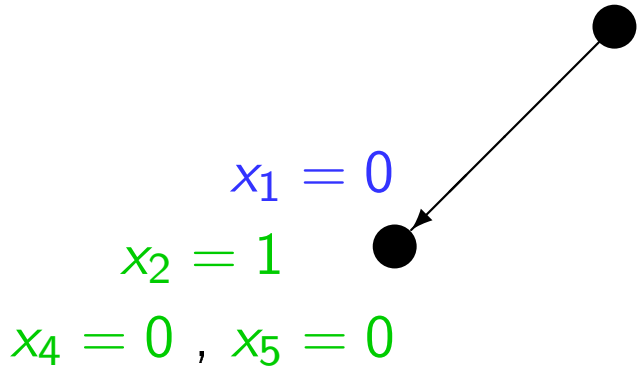
$$(\overline{1} \vee x_4 \vee x_5) \wedge \quad (\overline{1} \vee \overline{x_4}) \wedge \quad (\overline{1} \vee \overline{x_5})$$

Задача SAT и способы ее решения



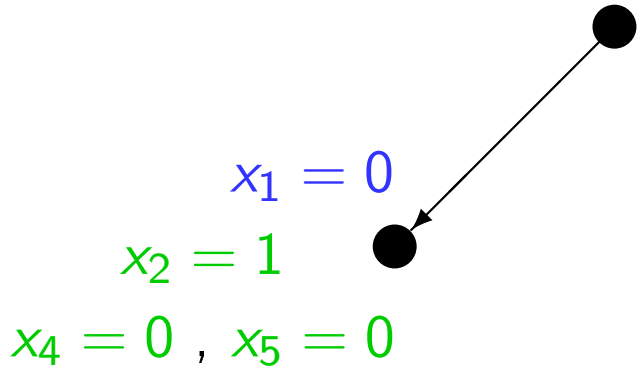
$(x_4 \vee x_5) \wedge$ $\overline{x_4} \wedge$ $\overline{x_5}$

Задача SAT и способы ее решения



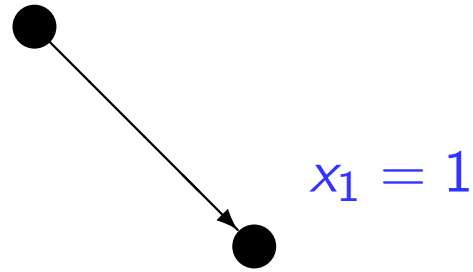
$(x_4 \vee x_5) \wedge$ $\overline{x_4} \wedge$ $\overline{x_5}$

Задача SAT и способы ее решения



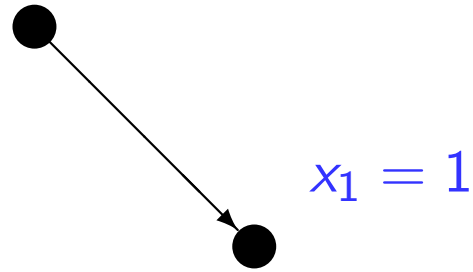
$(0 \vee 0)$

Задача SAT и способы ее решения



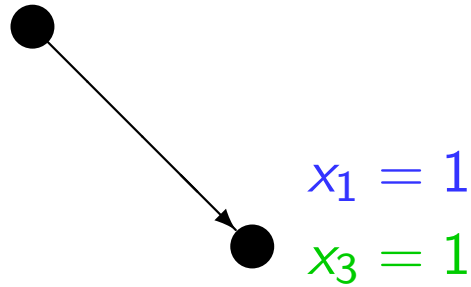
$$(x_1 \vee x_2) \wedge (\overline{x_1} \vee x_3) \wedge (\overline{x_2} \vee x_4 \vee x_5) \wedge (x_1 \vee \overline{x_2} \vee \overline{x_4}) \wedge (\overline{x_2} \vee \overline{x_5})$$

Задача SAT и способы ее решения



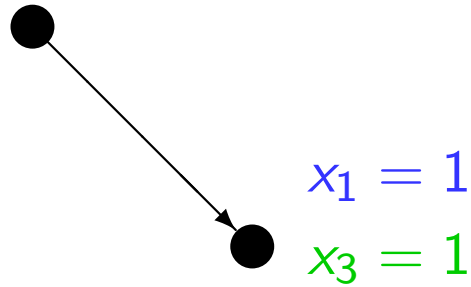
$$(1 \vee x_2) \wedge \ (\overline{1} \vee x_3) \wedge \ (\overline{x_2} \vee x_4 \vee x_5) \wedge (1 \vee \overline{x_2} \vee \overline{x_4}) \wedge \ (\overline{x_2} \vee \overline{x_5})$$

Задача SAT и способы ее решения



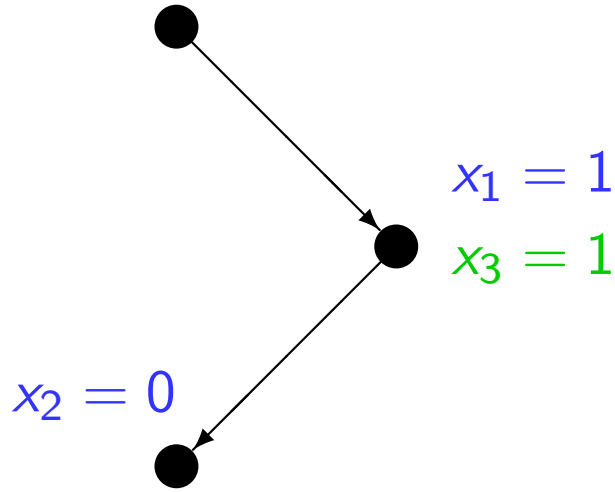
$x_3 \wedge (\overline{x_2} \vee x_4 \vee x_5) \wedge$ $(\overline{x_2} \vee \overline{x_5})$

Задача SAT и способы ее решения



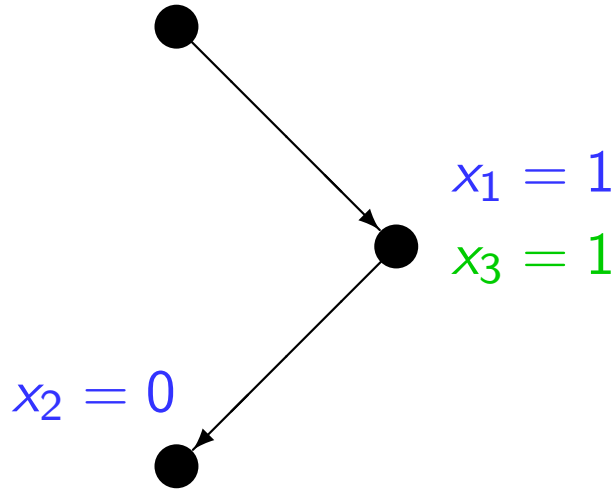
$(\overline{x_2} \vee x_4 \vee x_5) \wedge$ $(\overline{x_2} \vee \overline{x_5})$

Задача SAT и способы ее решения



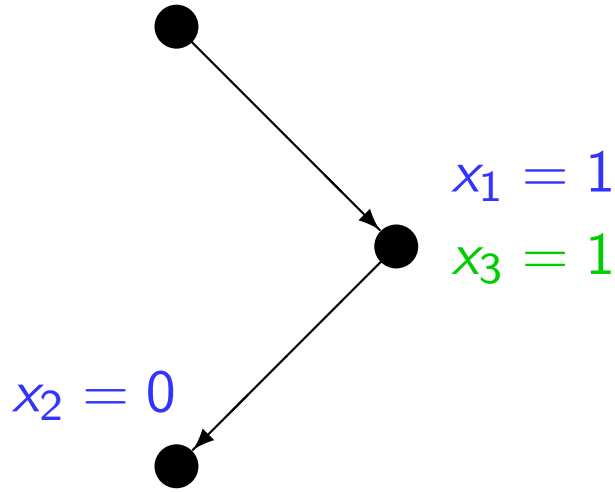
$(\overline{x_2} \vee x_4 \vee x_5) \wedge$ $(\overline{x_2} \vee \overline{x_5})$

Задача SAT и способы ее решения



$(\overline{0} \vee x_4 \vee x_5) \wedge$ $(\overline{0} \vee \overline{x_5})$

Задача SAT и способы ее решения



Задача SAT и способы ее решения

Современные инструментальные средства решения задачи SAT

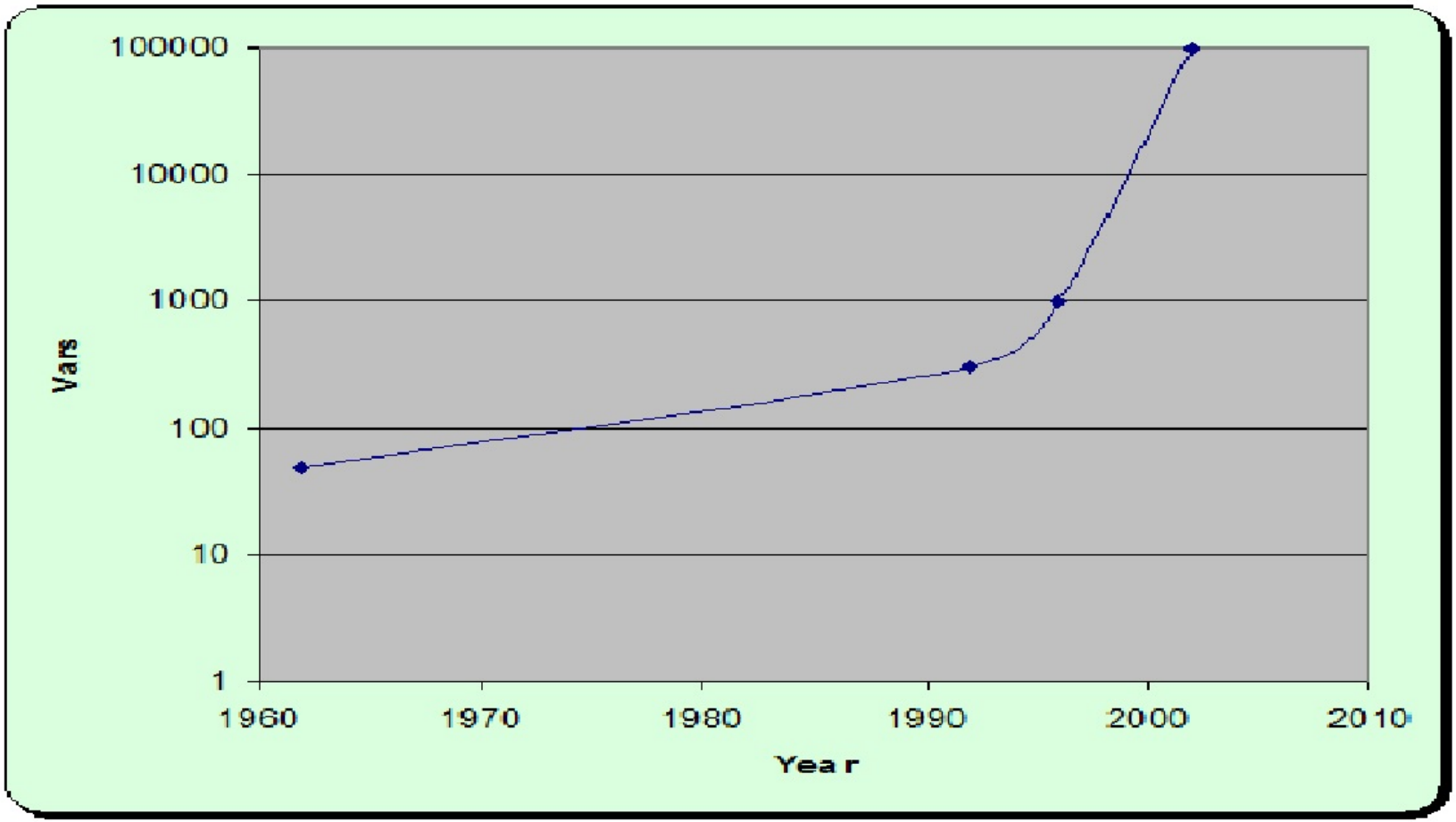
MiniSAT, BoolSAT, ArgoSAT, BCSAT, etc.

комбинируют основной алгоритм (DPLL, random walking) и несколько приемов, сокращающих перебор, — обнаружение и регистрация конфликтов, нехронологический откат, случайный перезапуск и др.

В результате постоянного и интенсивного совершенствования эти средства способны проверять выполнимость 3-КНФ, содержащих миллионы дизъюнктов и сотни тысяч переменных.

Задача SAT и способы ее решения

Стремительный прогресс в решении задачи SAT



Задача ограниченной верификации моделей программ (BMC)

Чтобы воспользоваться SAT-решателями, нужно уметь сводить исследуемую задачу к задаче SAT, т.е. построить такой транслятор $Tr : Problems \to 3\text{-КНФ}$, чтобы для любой конкретной задачи P из класса $Problems$ было верно

P имеет положительное решение $\Leftrightarrow$ $Tr(P)$ — выполнимая 3-КНФ

Задача ограниченной верификации моделей программ (BMC)

Чтобы воспользоваться SAT-решателями, нужно уметь сводить исследуемую задачу к задаче SAT, т.е. построить такой транслятор $Tr : Problems \to 3\text{-КНФ}$, чтобы для любой конкретной задачи P из класса $Problems$ было верно

P имеет положительное решение $\Leftrightarrow$ $Tr(P)$ — выполнимая 3-КНФ

Известно, что задачу model checking для PLTL невозможно эффективно свести к задаче SAT (почему?).

Задача ограниченной верификации моделей программ (BMC)

Чтобы воспользоваться SAT-решателями, нужно уметь сводить исследуемую задачу к задаче SAT, т.е. построить такой транслятор $Tr : Problems \to 3\text{-КНФ}$, чтобы для любой конкретной задачи P из класса $Problems$ было верно

P имеет положительное решение $\Leftrightarrow$ $Tr(P)$ — выполнимая 3-КНФ

Известно, что задачу model checking для PLTL невозможно эффективно свести к задаче SAT (почему?).

Поэтому приходится ограничиваться упрощенным вариантом задачи model checking — bounded model checking (BMC) :

- проверке подлежат только формулы вида $\mathbf{E}\varphi$, где φ — формула пути в позитивной нормальной форме (отрицания применяются только к элементарным высказываниям);
- проверка проводится только на начальных отрезках трасс фиксированной длины k в модели M .

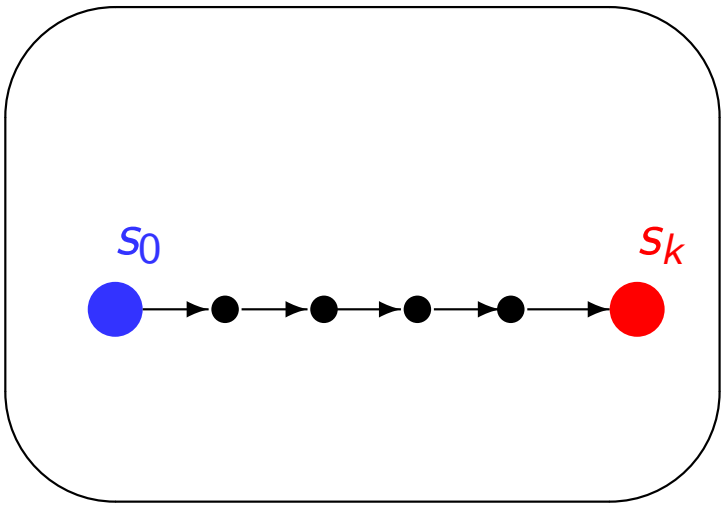
$$M \models_k \mathbf{E}\varphi$$

Задача ограниченной верификации моделей программ (BMC)

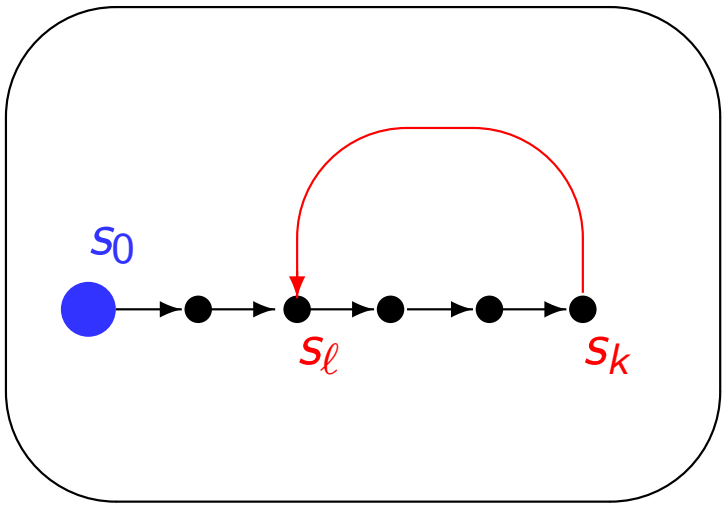
Мы будем рассматривать конечные пути двух типов, порожденные конечными последовательностями состояний

$$s_0, s_1, \ldots, s_k$$

k-путь



(ℓ, k)-путь



Если $\pi = s_0, s_1, \ldots, s_\ell, \ldots, s_k$ — это (ℓ, k)-путь, то запись $\widehat{\pi}$ будет обозначать бесконечный путь

$$\widehat{\pi} = s_0, s_1, \ldots, s_\ell, s_{\ell+1}, \ldots, s_k, s_\ell, s_{\ell+1}, \ldots, s_k, \ldots$$

Задача ограниченной верификации моделей программ (BMC)

Определение ограниченной выполнимости $\pi \models_k \varphi$

1. Если π — это (ℓ, k) -путь, то $\pi \models_k \varphi \Leftrightarrow \widehat{\pi} \models \varphi$.

Задача ограниченной верификации моделей программ (BMC)

Определение ограниченной выполнимости $\pi \models_k \varphi$

1. Если π — это (ℓ, k) -путь, то $\pi \models_k \varphi \Leftrightarrow \widehat{\pi} \models \varphi$.
2. Если π — это k -путь, то $\pi \models_k \varphi \Leftrightarrow \widehat{\pi} \models_k^0 \varphi$,
   где для любой пары $0 \le i \le k$
   - $\pi \models_k^i p \iff p \in L(s_i)$, $\pi \models_k^i \neg p \iff p \notin L(s_i)$,
   - $\pi \models_k^i f \wedge g \iff \pi \models_k^i f$ и $\pi \models_k^i f$,
   - $\pi \models_k^i f \vee g \iff \pi \models_k^i f$ или $\pi \models_k^i f$,
   - $\pi \models_k^i \mathbf{X} f \iff \pi \models_k^{i+1} f$ и $i < k$,
   - $\pi \models_k^i \mathbf{F} f \iff \pi \models_k^j f$ для некоторого $j: \ i \le j \le k$,
   - $\pi \not\models_k^i \mathbf{G} f$ для любых i, k,
   - $\pi \models_k^i f \, \mathbf{U} \, g \iff$ аналогично, самостоятельно ,
   - $\pi \models_k^i f \, \mathbf{R} \, g \iff$ аналогично, самостоятельно ,

Задача ограниченной верификации моделей программ (BMC)

Обратите внимание, что $\pi \models_k \neg \mathbf{G}\, f \not\iff \pi \models_k \mathbf{F} \neg f$

Задача 1.

А какие еще равносильности, выполняюшиеся в PLTL, нарушаются при переходе к отношению ограниченной выполнимости?
Действуют ли для отношения ограниченной выполнимости законы неподвижной точки?

Задача ограниченной верификации моделей программ (BMC)

Обратите внимание, что $\pi \models_k \neg \mathbf{G}\, f \not\iff \pi \models_k \mathbf{F} \neg f$

Задача 1.

А какие еще равносильности, выполняюшиеся в PLTL, нарушаются при переходе к отношению ограниченной выполнимости?
Действуют ли для отношения ограниченной выполнимости законы неподвижной точки?

Пусть φ — произвольная формула пути. Условимся использовать запись $M \models_k \mathbf{E}\, \varphi$ для обозначения заявления о том, что в модели M существует начальный k-путь или (ℓ, k)-путь π , для которого верно соотношение $\pi \models_k \varphi$.

Задача ограниченной верификации моделей программ (BMC)

Утверждение 1.

Для любой формулы пути в позитивной форме φ , для любого $k, k \geq 1$, и для любого (бесконечного) пути $\pi_\infty = \pi\pi'$, где π — k-путь, верно соотношение $\pi \models_k \varphi \Rightarrow \pi_\infty \models \varphi$.

Задача ограниченной верификации моделей программ (BMC)

Утверждение 1.

Для любой формулы пути в позитивной форме φ , для любого $k, k \geq 1$, и для любого (бесконечного) пути $\pi_\infty = \pi\pi'$, где π — k-путь, верно соотношение $\pi \models_k \varphi \Rightarrow \pi_\infty \models \varphi$.

Утверждение 2.

Пусть φ — формула пути в позитивной форме, и M — модель Крипке. Тогда если $M \not\models \mathbf{A}\neg\varphi$, то существует такой начальный k-путь или (ℓ, k)-путь π , для которого верно соотношение $\pi \models_k \varphi$.

Задача ограниченной верификации моделей программ (BMC)

Утверждение 1.

Для любой формулы пути в позитивной форме φ , для любого $k, k \geq 1$, и для любого (бесконечного) пути $\pi_\infty = \pi\pi'$, где π — k-путь, верно соотношение $\pi \models_k \varphi \;\Rightarrow\; \pi_\infty \models \varphi$.

Утверждение 2.

Пусть φ — формула пути в позитивной форме, и M — модель Крипке. Тогда если $M \not\models \mathbf{A}\neg\varphi$, то существует такой начальный k-путь или (ℓ, k)-путь π , для которого верно соотношение $\pi \models_k \varphi$.

Теорема 1.

Пусть φ — формула пути в позитивной форме, и M — модель Крипке. Тогда $M \not\models \mathbf{A}\neg\varphi$ в том и только том случае, когда для некоторого k верно соотношение $M \models_k \mathbf{E}\varphi$.

Задача ограниченной верификации моделей программ (BMC)

Таким образом, мы приходим к задаче BMC:

для заданных формулы пути в позитивной форме φ , модели Крипке M и целого k проверить соотношение $M \models_k \mathbf{E}\,\varphi$.

Покажем, как можно свести задачу BMC к задаче SAT.

Трансляция задачи BMC в задачу SAT

Для этого опишем алгоритм (транслятор), который для произвольной формулы пути φ , модели Крипке M и целого k строит такую булеву формулу $Trans[\varphi, M, k]$, которая выполнима в том и только том случае, когда $M \models_k \mathbf{E}\,\varphi$.

Трансляция задачи BMC в задачу SAT

Для этого опишем алгоритм (транслятор), который для произвольной формулы пути φ , модели Крипке M и целого k строит такую булеву формулу $Trans[\varphi, M, k]$, которая выполнима в том и только том случае, когда $M \models_k \mathbf{E}\,\varphi$.

Трансляция проводится в три этапа.

1). Вначале построим булеву формулу $PH[M, k]$, которая выполняется только на тех наборах значений переменных, которые представляют всевозможные начальные пути π длины k в модели M .

2) Затем построим булевы формулы $LP[k, \ell]$ и $NL[k]$, выполнимость которых служит индикатором существования/отсутствия цикла в (ℓ, k)-пути π .

3) В заключение построим булеву формулу $CH[i, k, \varphi]$, выполнимость которой является признаком того, что $\pi \models_k^i \varphi$.

Трансляция задачи BMC в задачу SAT

Заметим, что модель $M = (S, S_0, R, L)$ может быть представлена так:

- множество состояний S — это множество двоичных наборов некоторой длины n ;
- множество начальных состояний S_0 описывается булевой формулой I : $I(s) = 1$ тогда и только тогда, когда $s \in S_0$;
- отношение переходов R описывается булевой формулой T : $T(s', s'') = 1$ тогда и только тогда, когда $(s', s'') \in R$;
- для каждого атомарного высказывания p есть булева формула P : $P(s) = 1$ тогда и только тогда, когда $p \in L(s)$

Трансляция задачи BMC в задачу SAT

Заметим, что модель $M = (S, S_0, R, L)$ может быть представлена так:

- множество состояний S — это множество двоичных наборов некоторой длины n ;
- множество начальных состояний S_0 описывается булевой формулой I : $I(s) = 1$ тогда и только тогда, когда $s \in S_0$;
- отношение переходов R описывается булевой формулой T : $T(s', s'') = 1$ тогда и только тогда, когда $(s', s'') \in R$;
- для каждого атомарного высказывания p есть булева формула P : $P(s) = 1$ тогда и только тогда, когда $p \in L(s)$

Тогда $PH[M, k](\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_k) = I(x_0) \wedge \bigwedge_{i=1}^{k} T(\mathbf{x}_{i-1}, \mathbf{x}_i)$

$LP[k, \ell] = T(\mathbf{x}_k, \mathbf{x}_\ell)$,

$NL[k] = \bigwedge_{i=0}^{k} \overline{LP[k, \ell]}$.

Трансляция задачи BMC в задачу SAT

Утверждение 3.

Для любой последовательности состояний $s_0, s_1, \ldots, s_k$ в модели M эта последовательность образует k-путь $s_0 \to s_1 \to \cdots \to s_k$ в том и только том случае, когда $PH[M,k](s_0, s_1, \ldots, s_k) = 1$.

Утверждение 4.

Для любой последовательности состояний $s_0, s_1, \ldots, s_k$ в модели M эта последовательность образует (ℓ,k)-путь $s_0 \to s_1 \to \cdots \to s_k \to s_\ell$ в том и только том случае, когда $PH[M,k,\ell](s_0, s_1, \ldots, s_k) \wedge LP[k,\ell](s_0, s_1, \ldots, s_k) = 1$.

Трансляция задачи BMC в задачу SAT

Для параметров k, ℓ и целого $i, 0 \leq i \leq k$ будем полагать

$$succ(i, \ell, k) = \begin{cases} i+1 \text{ если } i < k, \\ \ell \text{ если } i = k. \end{cases}$$

Тогда для случая (ℓ, k)-пути π трансляция условия выполнимости $\pi \models_k^i \varphi$ такова:

- $CH[i, \ell, k, p] = P(\mathbf{x}_i)$ для атомарной формулы p ;
- $CH[i, \ell, k, \neg p] = \overline{P(\mathbf{x}_i)}$;
- $CH[i, \ell, k, f \wedge g] = CH[i, \ell, k, f] \wedge CH[i, \ell, k, g]$;
- $CH[i, \ell, k, f \vee g] = CH[i, \ell, k, f] \vee CH[i, \ell, k, g]$;
- $CH[i, \ell, k, \mathbf{X}\, f] = CH[succ(i, \ell, k), \ell, k, f]$;
- $CH[i, \ell, k, \mathbf{F}\, f] = \bigvee\limits_{j=min(i,\ell)}^{k} CH[j, \ell, k, f]$;
- $CH[i, \ell, k, \mathbf{G}\, f] = \bigwedge\limits_{j=min(i,\ell)}^{k} CH[i, \ell, k, f]$;

Трансляция задачи BMC в задачу SAT

Замечание 3.

Обратите внимание, что правила трансляции предусматривают введение вспомогательных булевых переменных.

Трансляция задачи BMC в задачу SAT

Замечание 3.

Обратите внимание, что правила трансляции предусматривают введение вспомогательных булевых переменных.

Например, для случая $CH[0, 0, 2, \mathbf{G}\,\mathbf{F}\,p]$ получаем

$$
\begin{aligned}
CH[0, 0, 2, \mathbf{G}\,\mathbf{F}\,p] \quad = \quad & (y_0 \ \wedge \ y_1 \ \wedge \ y_2) \wedge \\
& \wedge \ (y_0 \equiv (P(s_0) \vee P(s_1) \vee P(s_2))) \wedge \\
& \wedge \ (y_1 \equiv (P(s_0) \vee P(s_1) \vee P(s_2))) \wedge \\
& \wedge \ (y_2 \equiv (P(s_0) \vee P(s_1) \vee P(s_2))) \wedge
\end{aligned}
$$

Трансляция задачи BMC в задачу SAT

Замечание 3.

Обратите внимание, что правила трансляции предусматривают введение вспомогательных булевых переменных.

Например, для случая $CH[0, 0, 2, \mathbf{G}\,\mathbf{F}\,p]$ получаем

$$\begin{aligned} CH[0, 0, 2, \mathbf{G}\,\mathbf{F}\,p] \quad = \quad & (y_0 \ \wedge \ y_1 \ \wedge \ y_2) \wedge \\ & \wedge \ (y_0 \equiv (P(s_0) \vee P(s_1) \vee P(s_2))) \wedge \\ & \wedge \ (y_1 \equiv (P(s_0) \vee P(s_1) \vee P(s_2))) \wedge \\ & \wedge \ (y_2 \equiv (P(s_0) \vee P(s_1) \vee P(s_2))) \wedge \end{aligned}$$

Задача 2.

Это не слишком экономная трансляция. А нельзя ли сделать ее более эффективно?

Трансляция задачи BMC в задачу SAT

Замечание 3.

Обратите внимание, что правила трансляции предусматривают введение вспомогательных булевых переменных.

Например, для случая $CH[0, 0, 2, \mathbf{G}\,\mathbf{F}\,p]$ получаем

$$\begin{aligned} CH[0, 0, 2, \mathbf{G}\,\mathbf{F}\,p] \quad = \quad & (y_0 \ \wedge \ y_1 \ \wedge \ y_2) \wedge \\ & \wedge \ (y_0 \equiv (P(s_0) \vee P(s_1) \vee P(s_2))) \wedge \\ & \wedge \ (y_1 \equiv (P(s_0) \vee P(s_1) \vee P(s_2))) \wedge \\ & \wedge \ (y_2 \equiv (P(s_0) \vee P(s_1) \vee P(s_2))) \wedge \end{aligned}$$

Задача 2.

Это не слишком экономная трансляция. А нельзя ли сделать ее более эффективно?

Задача 3.

А как устроена трансляция формулы $CH[i, \ell, k, f\,\mathbf{U}\,g]$?

Трансляция задачи BMC в задачу SAT

Для случая k-пути π трансляция условия выполнимости $\pi \models_k^i \varphi$ такова:

- $CH[i,k,p] = P(\mathbf{x}_i)$ для атомарной формулы p ;
- $CH[i,k,\neg p] = \overline{P(\mathbf{x}_i)}$;
- $CH[i,k,f \wedge g] = CH[i,k,f] \wedge CH[i,k,g]$;
- $CH[i,k,f \vee g] = CH[i,k,f] \vee CH[i,k,g]$;
- $CH[i,k,\mathbf{X}\, f] = CH[i+1,k,k,f]$;
- $CH[i,k,\mathbf{F}\, f] = CH[i,k,f] \vee CH[i+1,k,\mathbf{F}\, f]$;
- $CH[i,k,\mathbf{G}\, f] = CH[i,k,f] \wedge CH[i+1,k,\mathbf{G}\, f]$;
- $CH[k+1,k,f] = 0$.

Трансляция задачи BMC в задачу SAT

Для случая k-пути π трансляция условия выполнимости $\pi \models_k^i \varphi$ такова:

- $CH[i,k,p] = P(\mathbf{x}_i)$ для атомарной формулы p ;
- $CH[i,k,\neg p] = \overline{P(\mathbf{x}_i)}$;
- $CH[i,k,f \wedge g] = CH[i,k,f] \wedge CH[i,k,g]$;
- $CH[i,k,f \vee g] = CH[i,k,f] \vee CH[i,k,g]$;
- $CH[i,k,\mathbf{X}\, f] = CH[i+1,k,k,f]$;
- $CH[i,k,\mathbf{F}\, f] = CH[i,k,f] \vee CH[i+1,k,\mathbf{F}\, f]$;
- $CH[i,k,\mathbf{G}\, f] = CH[i,k,f] \wedge CH[i+1,k,\mathbf{G}\, f]$;
- $CH[k+1,k,f] = 0$.

Задача 4.

А как устроена трансляция формулы $CH[i,k,f\ \mathbf{U}\ g]$?

Трансляция задачи BMC в задачу SAT

Окончательно имеем

$$Tr[\varphi,M,k] = Path[M,k] \wedge \Big(\underbrace{(NL[k] \wedge CH[0,k,\varphi])}_{k-\text{пути}} \vee \underbrace{\bigvee_{\ell=0}^{k} (LP[\ell,k] \wedge CH[0,\ell,k,\varphi])}_{(\ell,k)-\text{пути}} \Big)$$

Трансляция задачи BMC в задачу SAT

Окончательно имеем

$$Tr[\varphi,M,k] = Path[M,k] \wedge \Big(\underbrace{(NL[k] \wedge CH[0,k,\varphi])}_{k-\text{пути}} \vee \underbrace{\bigvee_{\ell=0}^{k} (LP[\ell,k] \wedge CH[0,\ell,k,\varphi])}_{(\ell,k)-\text{пути}} \Big)$$

Теорема 2.

Для любой формулы пути в позитивной форме φ , для любого k , и для любой модели Крипке M верно

$$M \models_k \mathbf{E}\,\varphi \iff Tr[\varphi,M,k] \; - \; \text{выполнимая булева формула.}$$

Трансляция задачи BMC в задачу SAT

А каков размер булевой формулы, который получается в результате такой трансляции BMC-to-SAT?

Трансляция задачи BMC в задачу SAT

А каков размер булевой формулы, который получается в результате такой трансляции BMC-to-SAT?

В том виде, в котором трансляция была описана здесь, булева формула $Tr[\varphi, M, k]$ имеет размер

$$O(k|M| + k^2|\varphi|).$$

Если воспользоваться более экономной трансляцией, то размер удается сократить до

$$O(k(|M| + |\varphi|)).$$

Трансляция задачи BMC в задачу SAT

Итак, размер проверяемой булевой формулы $O(k(|M| + |\varphi|))$.

Трансляция задачи BMC в задачу SAT

Итак, размер проверяемой булевой формулы $O(k(|M| + |\varphi|))$.

Однако для гарантированной проверки $M \models \mathbf{E}\,\varphi$ нужно использовать значение k равное **диаметру** $d(M)$ модели M .
В общем случае, $d(M) = 2^{O(|M|)}$.

Трансляция задачи BMC в задачу SAT

Итак, размер проверяемой булевой формулы $O(k(|M| + |\varphi|))$.

Однако для гарантированной проверки $M \models \mathbf{E}\,\varphi$ нужно использовать значение k равное **диаметру** $d(M)$ модели M . В общем случае, $d(M) = 2^{O(|M|)}$.

Кроме того, проверка выполнимости булевой формулы проводится в худшем случае за время, экспоненциально зависящее от размера этой формулы.

Таким образом, в худшем случае проверка $M \models \mathbf{E}\,\varphi$ посредством трансляции BMC-to-SAT проводится за время

$$2^{2^{O(|M|)}}.$$

Трансляция задачи BMC в задачу SAT

Итак, размер проверяемой булевой формулы $O(k(|M| + |\varphi|))$.

Однако для гарантированной проверки $M \models \mathbf{E}\,\varphi$ нужно использовать значение k равное **диаметру** $d(M)$ модели M .
В общем случае, $d(M) = 2^{O(|M|)}$.

Кроме того, проверка выполнимости булевой формулы проводится в худшем случае за время, экспоненциально зависящее от размера этой формулы.

Таким образом, в худшем случае проверка $M \models \mathbf{E}\,\varphi$ посредством трансляции BMC-to-SAT проводится за время

$$2^{2^{O(|M|)}}.$$

Для справки: табличный алгоритм верификации $M \models \mathbf{E}\,\varphi$ имеет сложность $2^{O(|M|+|\varphi|)}$.

В чем же гешефт?

Трансляция задачи BMC в задачу SAT

Преимущества BMC таковы:

1. BMC работает с неявным представлением модели;
2. подавляющее большинство ошибок располагаются «неглубоко»;
3. у многих практически важных моделей диаметр $d(M)$ невелик;

Трансляция задачи BMC в задачу SAT

Преимущества BMC таковы:

1. BMC работает с неявным представлением модели;
2. подавляющее большинство ошибок располагаются «неглубоко»;
3. у многих практически важных моделей диаметр $d(M)$ невелик;
4. **современные SAT-решатели неправдоподобно производительны на практике!**

Трансляция задачи BMC в задачу SAT

Практические преимущества BMC по сравнению с верификацией ROBDD:

Model	k	BDD	BMC
Circuit 1	5	114	2.5
Circuit 2	7	2	0.8
Circuit 3	7	106	2
Circuit 4	11	6189	2
Circuit 5	11	4196	10
Circuit 6	20	2795	236
Circuit 7	28	*	46
Circuit 8	10	5524	378
Circuit 9	37	*	195
Circuit 10	12	*	1070

Рис.: Результаты верификации (сек) различных проектов элетронных схем Intel при помощи средства верификации с использованием ROBDD (Forcast) и BMC (SAT-решатель Thunder)

КОНЕЦ ЛЕКЦИИ 11.