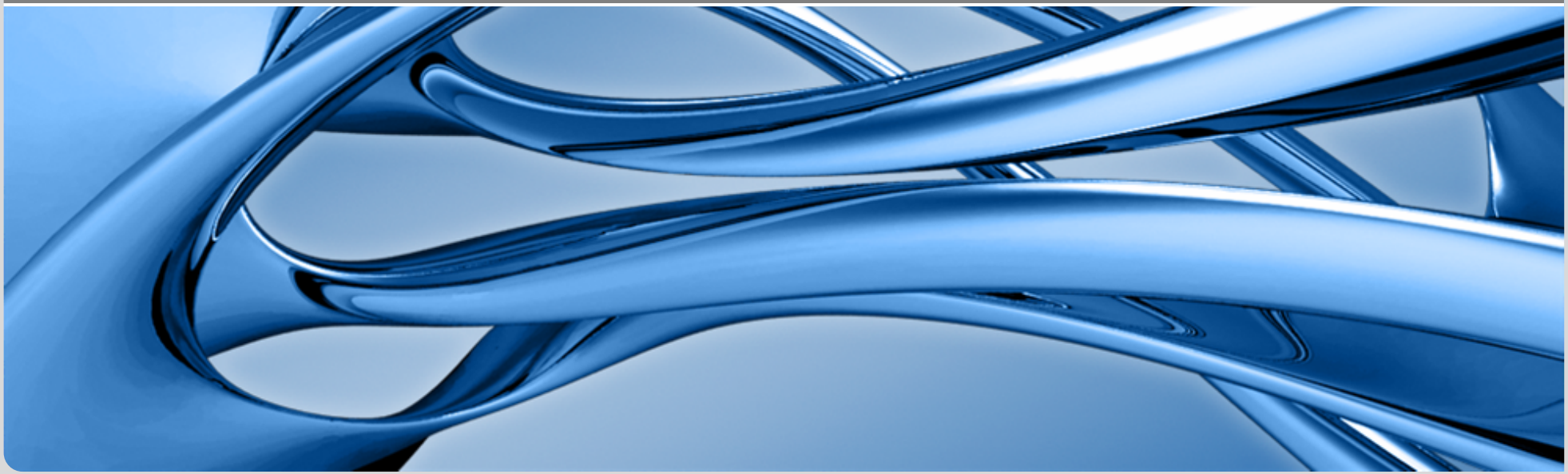
Karlsruher Institut für Technologie

SMT und DPLL(*T*)

Entscheidungsverfahren mit Anwendungen in der Softwareverifikation

STEPHAN FALKE — INSTITUT FÜR THEORETISCHE INFORMATIK (ITI)



KIT – Universität des Landes Baden-Württemberg und
nationales Forschungszentrum in der Helmholtz-Gemeinschaft

www.kit.edu

Inhalt

Abstraktes DPLL

- Regel-basierte Formulierung von DPLL
  - Sauber
  - Flexibel
  - Einfach
- Zustände:
  - *Fail*
  - $M \parallel F$
    - F: aussagenlogische Formel in CNF
    - M: Liste von Literalen
- M wird als partielle Zuweisung $M : \mathrm{Var}(F) \rightsquigarrow \{\top, \bot\}$ interpretiert

Inferenzregeln 1

UnitPropagate

$$M \parallel F, C \vee l \quad \Longrightarrow \quad M\ l \parallel F, C \vee l \quad \text{falls} \begin{cases} M \models \neg C \\ l \text{ undefiniert in } M \end{cases}$$

Decide

$$M \parallel F \quad \Longrightarrow \quad M\ l^{\mathrm{d}} \parallel F \quad \text{falls} \begin{cases} l \text{ oder } \neg l \text{ kommt in } F \text{ vor} \\ l \text{ undefiniert in } M \end{cases}$$

Inferenzregeln 2

Fail

$$M \parallel F, C \quad \Longrightarrow \quad \mathit{Fail} \qquad \text{falls} \begin{cases} M \models \neg C \\ M \text{ enthält kein Literal} \\ \quad \text{der Form } l^{\mathrm{d}} \end{cases}$$

Backtrack

$$M\ l^{\mathrm{d}}\ N \parallel F, C \quad \Longrightarrow \quad M\ \neg l \parallel F, C \qquad \text{falls} \begin{cases} M\ l^{\mathrm{d}}\ N \models \neg C \\ N \text{ enthält kein Literal} \\ \quad \text{der Form } k^{\mathrm{d}} \end{cases}$$

Beispiel

$$
\begin{aligned}
\emptyset &\parallel \neg A \vee \neg B,\ B \vee C,\ \neg A \vee \neg C \vee D,\ B \vee \neg C \vee \neg D,\ A \vee D && \Longrightarrow \text{(Decide)}\\
A^{\mathrm{d}} &\parallel \neg A \vee \neg B,\ B \vee C,\ \neg A \vee \neg C \vee D,\ B \vee \neg C \vee \neg D,\ A \vee D && \Longrightarrow \text{(UnitProp)}\\
A^{\mathrm{d}}\ \neg B &\parallel \neg A \vee \neg B,\ B \vee C,\ \neg A \vee \neg C \vee D,\ B \vee \neg C \vee \neg D,\ A \vee D && \Longrightarrow \text{(UnitProp)}\\
A^{\mathrm{d}}\ \neg B\ C &\parallel \neg A \vee \neg B,\ B \vee C,\ \neg A \vee \neg C \vee D,\ B \vee \neg C \vee \neg D,\ A \vee D && \Longrightarrow \text{(UnitProp)}\\
A^{\mathrm{d}}\ \neg B\ C\ D &\parallel \neg A \vee \neg B,\ B \vee C,\ \neg A \vee \neg C \vee D,\ B \vee \neg C \vee \neg D,\ A \vee D && \Longrightarrow \text{(Backtrack)}\\
\neg A &\parallel \neg A \vee \neg B,\ B \vee C,\ \neg A \vee \neg C \vee D,\ B \vee \neg C \vee \neg D,\ A \vee D && \Longrightarrow \text{(UnitProp)}\\
\neg A\ D &\parallel \neg A \vee \neg B,\ B \vee C,\ \neg A \vee \neg C \vee D,\ B \vee \neg C \vee \neg D,\ A \vee D && \Longrightarrow \text{(Decide)}\\
\neg A\ D\ \neg C^{\mathrm{d}} &\parallel \neg A \vee \neg B,\ B \vee C,\ \neg A \vee \neg C \vee D,\ B \vee \neg C \vee \neg D,\ A \vee D && \Longrightarrow \text{(UnitProp)}\\
\neg A\ D\ \neg C^{\mathrm{d}}\ B &\parallel \neg A \vee \neg B,\ B \vee C,\ \neg A \vee \neg C \vee D,\ B \vee \neg C \vee \neg D,\ A \vee D && \not\Longrightarrow
\end{aligned}
$$

$A = \bot,\ B = \top,\ C = \bot,\ D = \top$ ist ein Modell der Formel

Eigenschaften

Satz

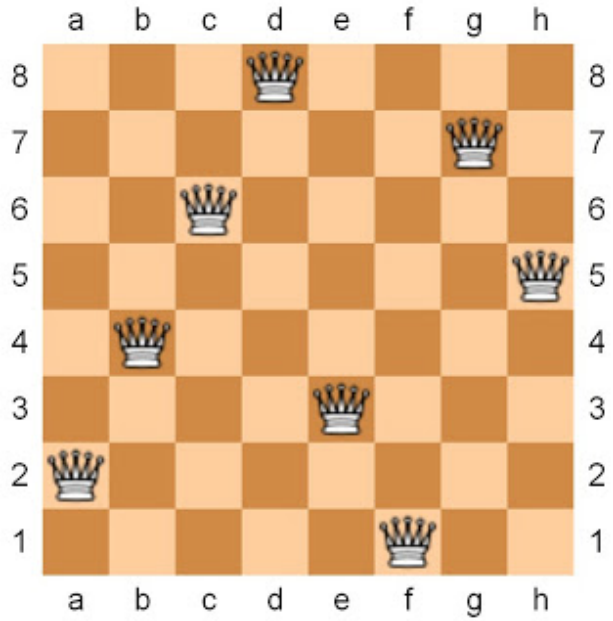
Alle Ableitungen $\emptyset \parallel F \implies S_1 \implies \ldots$ sind endlich

Satz

Sei $\emptyset \parallel F \implies S_1 \implies \ldots \implies S$ eine maximale Ableitung

- F ist unerfüllbar gdw. $S = \mathit{Fail}$
- Falls S die Form $M \parallel F$ hat dann ist M ein Modell von F

Das *n*-Damen Problem



Platziere n Damen so auf einem $n \times n$-Schachbrett dass keine Dame eine andere Dame schlagen kann

Lösung mittels Logik

- Führe eine 0/1-Variable pro Feld ein

$$\begin{matrix} X_{1,1} & \cdots & X_{1,N} \\ \vdots & & \vdots \\ X_{N,1} & \cdots & X_{N,N} \end{matrix}$$

- Genau eine Dame pro Zeile/Spalte:

$$\sum_{j=1}^{N} X_{1,j} = 1 \wedge \ldots \wedge \sum_{j=1}^{N} X_{N,j} = 1 \wedge \sum_{i=1}^{N} X_{i,1} = 1 \wedge \ldots \wedge \sum_{i=1}^{N} X_{i,N} = 1$$

- Höchstens eine Dame pro Diagonale:

$$\begin{aligned} X_{1,1} + X_{2,2} + \ldots + X_{N,N} &\leq 1 \\ X_{1,2} + X_{2,3} + \ldots + X_{N-1,N} &\leq 1 \\ X_{1,3} + X_{2,4} + \ldots + X_{N-2,N} &\leq 1 \\ &\vdots \end{aligned}$$

SAT?

- Logik benutzt Schreibweisen aus der Arithmetik
- Diese müssen nach SAT kodiert werden
- $\sum_{j=1}^{N} X_{1,j} = 1$ kann kodiert werden als

$$(X_{1,1} \vee \ldots \vee X_{1,N}) \wedge \neg(X_{1,1} \wedge X_{1,2}) \wedge \neg(X_{1,1} \wedge X_{1,3}) \wedge \ldots \wedge \neg(X_{1,N-1} \wedge X_{1,N})$$

- $\sum_{i=1}^{N} X_{i,i} \leq 1$ kann kodiert werden als

$$\neg(X_{1,1} \wedge X_{2,2}) \wedge \neg(X_{1,1} \wedge X_{3,3}) \wedge \ldots \wedge \neg(X_{N-1,N-1} \wedge X_{N,N})$$

- Beobachtung: Aussagenlogik ist wie Maschinensprache

SMT!

- Wunsch: Logik, die Arithmetik beherrscht
- n-Damen Problem benötigt dann (fast) keine weitere Kodierung
  - $X_{i,j} = 0 \vee X_{i,j} = 1$ für alle $X_{i,j}$
- Wunsch: Unterstützung weiterer eingebaute Datentypen
- Lösung: Benutze SMT (satisfiability modulo theories)!
- Beobachtung: SMT ist wie Haskell/OCaml/Scala/...

Theorien 1

Definition

Eine Theorie T besteht aus

- einer Signatur Σ bestehend aus Konstanten-, Funktions-, und Prädikatensymbolen
- einer Menge Ax von Axiomen (Sätze der Prädikatenlogik)

- Die Symbole in Σ haben keine festgelegte Bedeutung
- Die Bedeutung wird erst durch Ax fest gelegt

Theorie der Uninterpretierten Funktionen

Beispiel

- $\Sigma = \{=, a, b, c, \ldots, f, g, h, \ldots\}$
- Axiome Ax:
  1. $\forall x.\ x = x$ (Reflexivität)
  2. $\forall x, y.\ x = y \rightarrow y = x$ (Symmetrie)
  3. $\forall x, y, z.\ x = y \wedge y = z \rightarrow x = z$ (Transitivität)
  4. für jedes n-stellige Funktionssymbol $f \in \Sigma$:
     $\forall x_1, \ldots, x_n, y_1, \ldots, y_n.\ \bigwedge_{i=1}^{n} x_i = y_i \rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n)$
     (Kongruenz)
- (Kongruenz) ist ein Axiomschema das unendliche viele Axiome repräsentiert

Arithmetische Theorien 1

Karlsruher Institut für Technologie

Beispiel (Peano Arithmetik)

- $\Sigma = \{=, 0, 1, +, *\}$
- Axiome Ax:
  1. Axiome der Theorie der Uninterpretierten Funktionen
  2. $\forall x.\ \neg(x + 1 = 0)$ (Null)
  3. $\forall x, y.\ x + 1 = y + 1 \rightarrow x = y$ (Nachfolger)
  4. $F[0] \wedge (\forall x.\ F[x] \rightarrow F[x + 1]) \rightarrow \forall x.\ F[x]$ (Induktion)
  5. $\forall x.\ x + 0 = x$ (Plus Null)
  6. $\forall x, y.\ x + (y + 1) = (x + y) + 1$ (Plus Nachfolger)
  7. $\forall x.\ x * 0 = 0$ (Multiplikation Null)
  8. $\forall x, y.\ x * (y + 1) = (x * y) + x$ (Multiplikation Nachfolger)
- (Induktion) ist ein Axiomschema

Arithmetische Theorien 2

Beispiel (Presburger Arithmetik)

- $\Sigma = \{=, 0, 1, +\}$ keine Multiplikation!
- Axiome *Ax*:
  1. Axiome der Theorie der Uninterpretierten Funktionen
  2. $\forall x.\ \neg(x + 1 = 0)$ (Null)
  3. $\forall x, y.\ x + 1 = y + 1 \rightarrow x = y$ (Nachfolger)
  4. $F[0] \wedge (\forall x.\ F[x] \rightarrow F[x + 1]) \rightarrow \forall x.\ F[x]$ (Induktion)
  5. $\forall x.\ x + 0 = x$ (Plus Null)
  6. $\forall x, y.\ x + (y + 1) = (x + y) + 1$ (Plus Nachfolger)
- (Induktion) ist ein Axiomschema

Beispiel (Lineare Arithmetik ganzer Zahlen)

- $\Sigma = \{=, >, +, -, \ldots, -2, -1, 0, 1, 2, \ldots\}$ keine Multiplikation!
- Axiome *Ax*: ...

Theorie der Arrays

Beispiel

- $\Sigma = \{=, \text{read}, \text{write}\}$
- Axiome Ax:

1. $\forall x.\ x = x$	(Reflexivität)
2. $\forall x, y.\ x = y \rightarrow y = x$	(Symmetrie)
3. $\forall x, y, z.\ x = y \wedge y = z \rightarrow x = z$	(Transitivität)
4. $\forall a, i, j.\ i = j \rightarrow \text{read}(a, i) = \text{read}(a, j)$	(Array Kongruenz)
5. $\forall a, v, i, j.\ i = j \rightarrow \text{read}(\text{write}(a, i, v), j) = v$	(read-über-write 1)
6. $\forall a, v, i, j.\ i \neq j \rightarrow \text{read}(\text{write}(a, i, v), j) = \text{read}(a, j)$	(read-über-write 2)
7. $\forall a, b.\ a = b \leftrightarrow (\forall i.\ \text{read}(a, i) = \text{read}(b, i))$	(Extensionalität)

Theorien 2

Definition
Eine Σ-Formel F ist T-erfüllbar gdw. $Ax \cup \{F\}$ erfüllbar ist

Definition
Eine Σ-Formel F ist T-gültig gdw. $Ax \models F$

Theorien 3

Definition
Eine Theorie T ist entscheidbar gdw. T-Erfüllbarkeit für jede Σ-Formel entschieden werden kann

Definition
Ein Fragment einer Theorie T ist eine syntaktisch eingeschränkte Teilmenge aller Σ-Formeln

- Beispiel: Quantoren-freies Fragment (QFF)

Definition
Eine Fragment einer Theorie T ist entscheidbar gdw. T-Erfüllbarkeit für jede Σ-Formel des Fragments entschieden werden kann

Entscheidbarkeit

Theorie	Entscheidbar?	QFF entscheidbar?
Uninterpretierte Funktionen	—	√
Peano Arithmetik	—	—
Presburger Arithmetik	√	√
Lineare Arithmetik ganzer Zahlen	√	√
Arrays	—	√

DPLL(T): Idee

- Im folgenden: nur quantoren-freie Σ-Formeln
- Benutze DPLL um Konjunktionen von T-Literalen zu erzeugen
  - diese T-Literale erfüllen das Boolesche Skelett der Formel
- Prüfe ob die Konjunktion konsistent in T ist
  - Falls ja, extrahiere ein T-Modell
  - Falls nein, schließe diesen “Kandidaten” aus

Beispiel

Lineare Arithmetik ganzer Zahlen:

$$y \geq 1 \wedge (x < 0 \vee y < 1) \wedge (x \geq 0 \vee y < 0)$$

Boolesches Skelett:

$$A \wedge (B \vee C) \wedge (D \vee E)$$

Modell für das Boolesche Skelett:

$$A,\ C,\ E$$

Entsprechende T-Literale:

$$y \geq 1,\ y < 1,\ y < 0$$

Inkonsistente Konjunktion von T-Literalen:

$$y \geq 1 \wedge y < 1 \wedge y < 0$$

Schließe diesen “Kandidaten” aus:

$$\neg(y \geq 1) \vee \neg(y < 1)$$

Beispiel

Lineare Arithmetik ganzer Zahlen:

$$y \geq 1 \wedge (x < 0 \vee y < 1) \wedge (x \geq 0 \vee y < 0) \wedge [\neg(y \geq 1) \vee \neg(y < 1)]$$

Boolesches Skelett:

$$A \wedge (B \vee C) \wedge (D \vee E) \wedge (\neg A \vee \neg C)$$

Modell für das Boolesche Skelett:

$$A,\ \neg C,\ B,\ D$$

Entsprechende T-Literale:

$$y \geq 1,\ \neg(y < 1),\ x < 0,\ x \geq 0$$

Inkonsistente Konjunktion von T-Literalen:

$$y \geq 1 \wedge \neg(y < 1) \wedge x < 0 \wedge x \geq 0$$

Schließe diesen “Kandidaten” aus:

$$\neg(x < 0) \vee \neg(x \geq 0)$$

Beispiel

Lineare Arithmetik ganzer Zahlen:

$$y \geq 1 \wedge (x < 0 \vee y < 1) \wedge (x \geq 0 \vee y < 0) \wedge [\neg(y \geq 1) \vee \neg(y < 1)] \wedge [\neg(x < 0) \vee \neg(x \geq 0)]$$

Boolesches Skelett:

$$A \wedge (B \vee C) \wedge (D \vee E) \wedge (\neg A \vee \neg C) \wedge (\neg B \vee \neg D)$$

Modell für das Boolesche Skelett:

$$A,\ \neg C,\ B,\ \neg D,\ E$$

Entsprechende T-Literale:

$$y \geq 1,\ \neg(y < 1),\ x < 0,\ \neg(x \geq 0),\ y < 0$$

Inkonsistente Konjunktion von T-Literalen:

$$y \geq 1 \wedge \neg(y < 1) \wedge x < 0 \wedge \neg(x \geq 0) \wedge y < 0$$

Schließe diesen “Kandidaten” aus:

$$\neg(y \geq 1) \vee \neg(y < 0)$$

Beispiel

Lineare Arithmetik ganzer Zahlen:

$$y \geq 1 \wedge (x < 0 \vee y < 1) \wedge (x \geq 0 \vee y < 0) \wedge [\neg(y \geq 1) \vee \neg(y < 1)] \wedge [\neg(x < 0) \vee \neg(x \geq 0)] \wedge [\neg(y \geq 1) \vee \neg(y < 0)]$$

Boolesches Skelett:

$$A \wedge (B \vee C) \wedge (D \vee E) \wedge (\neg A \vee \neg C) \wedge (\neg B \vee \neg D) \wedge (\neg A \vee \neg E)$$

Boolesches Skelett hat kein Modell

$\Longrightarrow$ Ursprüngliche Formel hat kein T-Modell

Abstraktes DPLL(T)

- Erweiterung von abstraktem DPLL
- Parametrisiert durch eine Theorie T
- Benötigt einen T-Solver der die Konsistenz einer Konjunktion von T-Literalen entscheiden kann
- Zustände:
  - *Fail*
  - $M \parallel F$
    - F: Quantorenfreie Σ-Formal in CNF
    - M: Liste von T-Literalen
- Konsistenz von M wird durch T-Solver überprüft

Inferenzregeln 1 (aus DPLL)

Karlsruher Institut für Technologie

UnitPropagate

$$M \parallel F, C \vee l \quad \Longrightarrow_T \quad M\, l \parallel F, C \vee l \quad \text{falls} \begin{cases} M \models \neg C \\ l \text{ undefiniert in } M \end{cases}$$

Decide

$$M \parallel F \quad \Longrightarrow_T \quad M\, l^{\mathrm{d}} \parallel F \quad \text{falls} \begin{cases} l \text{ oder } \neg l \text{ kommt in } F \text{ vor} \\ l \text{ undefiniert in } M \end{cases}$$

Inferenzregeln 2 (aus DPLL)

Karlsruher Institut für Technologie

Fail

$$M \parallel F, C \quad \Longrightarrow_T \quad \mathit{Fail} \qquad \text{falls} \begin{cases} M \models \neg C \\ M \text{ enthält kein Literal} \\ \quad \text{der Form } l^{\mathrm{d}} \end{cases}$$

Backtrack

$$M\ l^{\mathrm{d}}\ N \parallel F, C \quad \Longrightarrow_T \quad M\ \neg l \parallel F, C \qquad \text{falls} \begin{cases} M\ l^{\mathrm{d}}\ N \models \neg C \\ N \text{ enthält kein Literal} \\ \quad \text{der Form } k^{\mathrm{d}} \end{cases}$$

Inferenzregeln 3 (neu!)

TheoryInconsistent

$$M \parallel F \quad \Longrightarrow_T \quad \emptyset \parallel F, \neg l_1 \vee \ldots \vee \neg l_n \quad \text{falls} \begin{cases} \{l_1, \ldots, l_n\} \subseteq M \\ \neg l_1 \vee \ldots \vee \neg l_n \ T\text{-gültig} \end{cases}$$

TheoryPropagate

$$M \parallel F \quad \Longrightarrow_T \quad M\ l \parallel F \qquad \text{falls} \begin{cases} M \models_T l \\ l \text{ oder } \neg l \text{ kommt} \\ \quad \text{in } F \text{ vor} \\ l \text{ undefiniert in } M \end{cases}$$

Eigenschaften

Satz
Alle Ableitungen $\emptyset \parallel F \implies_T S_1 \implies_T \ldots$ sind endlich

Satz
Sei $\emptyset \parallel F \implies_T S_1 \implies_T \ldots \implies_T S$ eine maximale Ableitung

- F ist unerfüllbar gdw. $S = \mathit{Fail}$
- Falls S die Form $M \parallel F$ hat dann hat M ein T-Modell und jedes T-Modell von M ist auch ein T-Modell von F

Ausblick

- T-Solver für:
  - Uninterpretierte Funktionen
  - Lineare Arithmetik ganzer Zahlen
  - Bitvektoren
  - Arrays
- Kombination von T-Solvern