

Künstliche Intelligenz

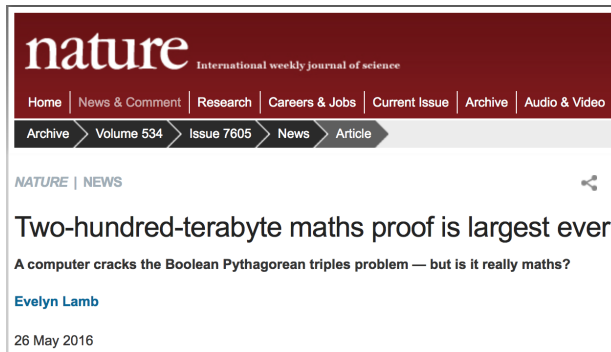
Prof. Dr. Christoph Benzmüller (& Prof. Dr. Christoph Schommer)
FU Berlin & U Luxembourg

Summer 2019

Topics addressed here:

- ▶ Knowledge Representation and Reasoning
- ▶ Propositional Logic and SAT Solving
- ▶ Davis-Putnam-Logemann-Loveland (DPLL) Algorithmus

Beispiel einer Erfolgsgeschichte



The screenshot shows the top section of the Nature journal website. The header is dark red with the 'nature' logo in white. Below the logo is the tagline 'International weekly journal of science'. A navigation bar contains links: Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. Below this is a secondary navigation bar with links: Archive, Volume 534, Issue 7605, News (highlighted), and Article. The main content area has a light gray background. On the left, it says 'NATURE | NEWS'. The article title is 'Two-hundred-terabyte maths proof is largest ever'. Below the title is a subtitle: 'A computer cracks the Boolean Pythagorean triples problem — but is it really maths?'. The author's name is 'Evelyn Lamb'. The date is '26 May 2016'. There is a share icon on the right side of the article title.

nature
International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive | Volume 534 | Issue 7605 | **News** | Article

NATURE | NEWS

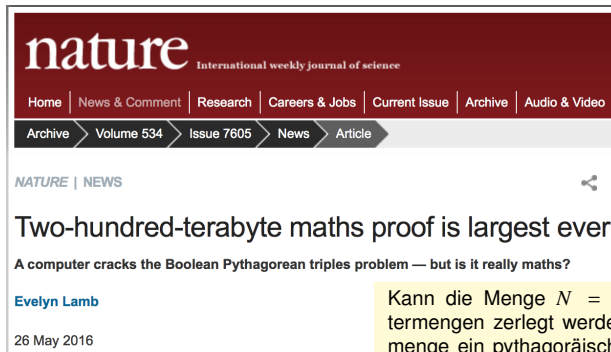
Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

Beitrag von
M. Heule
O. Kullmann
V. Marek



The screenshot shows the top navigation bar of the Nature journal website. The header is dark red with the 'nature' logo in white. Below the logo, it says 'International weekly journal of science'. The navigation bar includes links for Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. A secondary bar highlights 'Archive', 'Volume 534', 'Issue 7605', 'News', and 'Article'. The main content area shows the article title 'Two-hundred-terabyte maths proof is largest ever' by Evelyn Lamb, dated 26 May 2016. A share icon is visible to the right of the article title.

nature
International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive > Volume 534 > Issue 7605 > News > Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

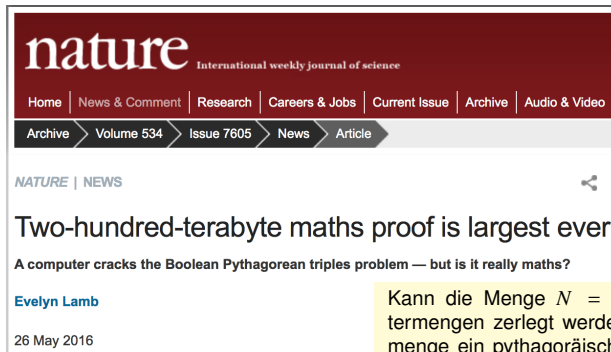
Beitrag von
M. Heule
O. Kullmann
V. Marek

Kann die Menge $N = \{1, 2, \dots, n\}$ in zwei Untermengen zerlegt werden, so dass keine Untermenge ein pythagoräisches Triple (a, b, c) enthält mit $a^2 + b^2 = c^2$?

$n = 10$

$$3^2 + 4^2 = 5^2$$

(wähle die Farbe der anderen Zahlen beliebig)



The screenshot shows the top section of the Nature journal website. The header is dark red with the 'nature' logo in white. Below the logo is the tagline 'International weekly journal of science'. A navigation bar contains links: Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. Below this is a secondary navigation bar with links: Archive, Volume 534, Issue 7605, News, and Article. The main content area has a sub-header 'NATURE | NEWS' and a share icon. The article title is 'Two-hundred-terabyte maths proof is largest ever'. Below the title is a subtitle 'A computer cracks the Boolean Pythagorean triples problem — but is it really maths?'. The author's name 'Evelyn Lamb' is listed, followed by the date '26 May 2016'.

nature
International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive | Volume 534 | Issue 7605 | News | Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

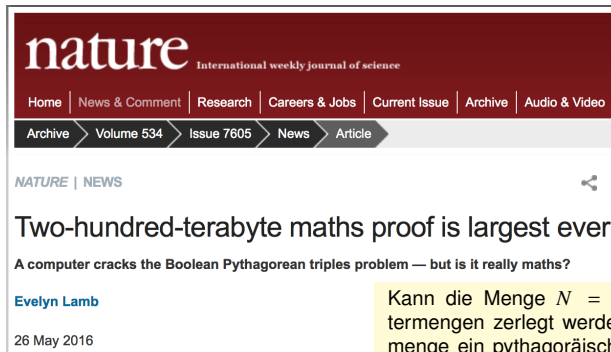
Beitrag von
M. Heule
O. Kullmann
V. Marek

Kann die Menge $N = \{1, 2, \dots, n\}$ in zwei Untermengen zerlegt werden, so dass keine Untermenge ein pythagoräisches Triple (a, b, c) enthält mit $a^2 + b^2 = c^2$?

$n = 10$

$$3^2 + 4^2 = 5^2$$

(wähle die Farbe der anderen Zahlen beliebig)



The screenshot shows the top navigation bar of the Nature journal website. The header is dark red with the 'nature' logo in white. Below the logo, it says 'International weekly journal of science'. The navigation bar includes links for Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. A secondary bar shows 'Archive', 'Volume 534', 'Issue 7605', 'News', and 'Article'. The main content area has a sub-header 'NATURE | NEWS' and a share icon. The article title is 'Two-hundred-terabyte maths proof is largest ever'. Below the title is a subtitle: 'A computer cracks the Boolean Pythagorean triples problem — but is it really maths?'. The author's name 'Evelyn Lamb' is listed, followed by the date '26 May 2016'.

nature
International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive > Volume 534 > Issue 7605 > News > Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

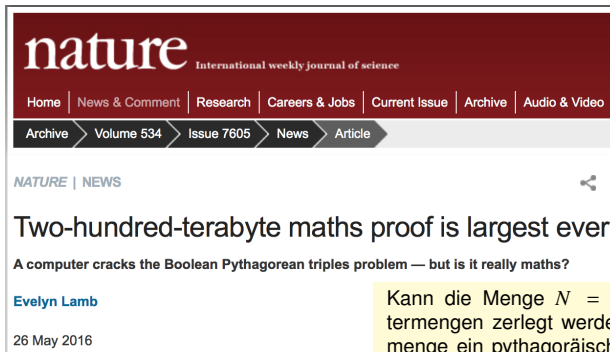
Beitrag von
M. Heule
O. Kullmann
V. Marek

Kann die Menge $N = \{1, 2, \dots, n\}$ in zwei Untermengen zerlegt werden, so dass keine Untermenge ein pythagoräisches Triple (a, b, c) enthält mit $a^2 + b^2 = c^2$?

$n = 10$

$$3^2 + 4^2 = 5^2$$

(wähle die Farbe der anderen Zahlen beliebig)



The screenshot shows the top navigation bar of the Nature journal website. The header is dark red with the 'nature' logo in white. Below the logo is the tagline 'International weekly journal of science'. A horizontal menu contains links: Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. Below this is a secondary navigation bar with links: Archive, Volume 534, Issue 7605, News, and Article. The 'News' link is highlighted. Below the navigation bar, the text 'NATURE | NEWS' is displayed. The main headline reads 'Two-hundred-terabyte maths proof is largest ever'. Below the headline is a sub-headline: 'A computer cracks the Boolean Pythagorean triples problem — but is it really maths?'. The author's name 'Evelyn Lamb' is listed below the sub-headline. The date '26 May 2016' is at the bottom left of the article preview.

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive | Volume 534 | Issue 7605 | News | Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

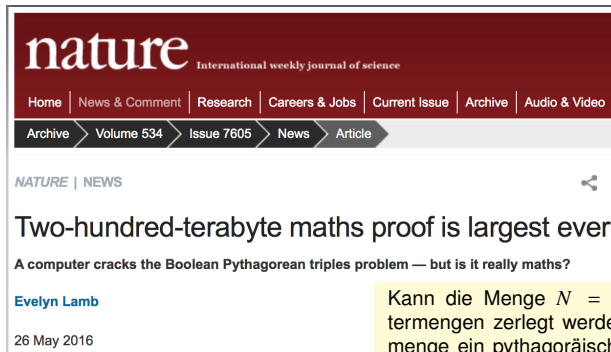
Beitrag von
M. Heule
O. Kullmann
V. Marek

Kann die Menge $N = \{1, 2, \dots, n\}$ in zwei Untermengen zerlegt werden, so dass keine Untermenge ein pythagoräisches Triple (a, b, c) enthält mit $a^2 + b^2 = c^2$?

$n = 10$

$$3^2 + 4^2 = 5^2$$

Verboten: nur eine Farbe!



The screenshot shows the top navigation bar of the Nature journal website. The header is dark red with the 'nature' logo in white. Below the logo, it says 'International weekly journal of science'. The navigation bar includes links for Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. A secondary bar highlights Archive, Volume 534, Issue 7605, News, and Article. The main content area shows the article title 'Two-hundred-terabyte maths proof is largest ever' by Evelyn Lamb, dated 26 May 2016. A share icon is visible to the right of the title.

nature
International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive | Volume 534 | Issue 7605 | News | Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

Beitrag von
M. Heule
O. Kullmann
V. Marek

Kann die Menge $N = \{1, 2, \dots, n\}$ in zwei Untermengen zerlegt werden, so dass keine Untermenge ein pythagoräisches Triple (a, b, c) enthält mit $a^2 + b^2 = c^2$?

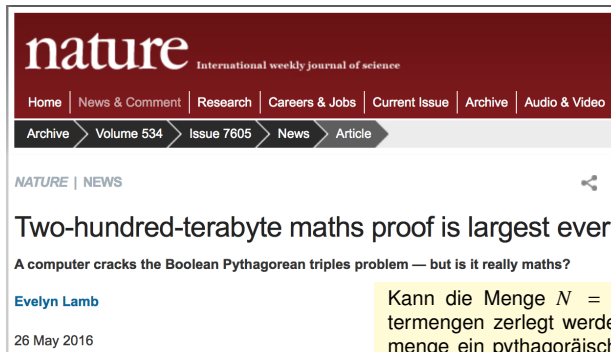
$n = 20$

$$3^2 + 4^2 = 5^2$$

$$5^2 + 12^2 = 13^2$$

$$8^2 + 15^2 = 17^2$$

(wähle die Farbe der anderen Zahlen beliebig)



The screenshot shows the top navigation bar of the Nature journal website. The header is dark red with the 'nature' logo in white. Below the logo, it says 'International weekly journal of science'. The navigation bar includes links for Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. A secondary bar highlights 'Archive', 'Volume 534', 'Issue 7605', 'News', and 'Article'. The main content area shows the article title 'Two-hundred-terabyte maths proof is largest ever' by Evelyn Lamb, dated 26 May 2016. A share icon is visible to the right of the title.

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive | Volume 534 | Issue 7605 | News | Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

Beitrag von
M. Heule
O. Kullmann
V. Marek

Kann die Menge $N = \{1, 2, \dots, n\}$ in zwei Untermengen zerlegt werden, so dass keine Untermenge ein pythagoräisches Triple (a, b, c) enthält mit $a^2 + b^2 = c^2$?

$n = 30$


$$3^2 + 4^2 = 5^2$$

$$5^2 + 12^2 = 13^2$$

$$8^2 + 15^2 = 17^2$$

$$7^2 + 24^2 = 25^2$$

(choose color of the other numbers arbitrarily)



International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive > Volume 534 > Issue 7605 > News > Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

Beitrag von
M. Heule
O. Kullmann
V. Marek

Kann die Menge $N = \{1, 2, \dots, n\}$ in zwei Untermengen zerlegt werden, so dass keine Untermenge ein pythagoräisches Triple (a, b, c) enthält mit $a^2 + b^2 = c^2$?

$n = 40$

$$3^2 + 4^2 = 5^2$$

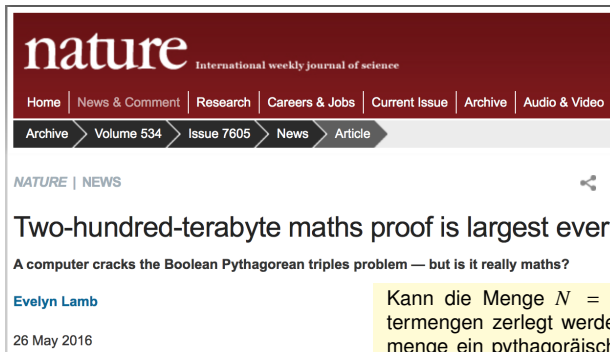
$$5^2 + 12^2 = 13^2$$

$$8^2 + 15^2 = 17^2$$

$$7^2 + 24^2 = 25^2$$

$$20^2 + 21^2 = 29^2$$

$$12^2 + 35^2 = 37^2$$



The screenshot shows the top navigation bar of the Nature journal website. The header is dark red with the 'nature' logo in white. Below the logo is the text 'International weekly journal of science'. The navigation bar includes links for Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. A secondary navigation bar below this highlights 'Archive', 'Volume 534', 'Issue 7605', 'News', and 'Article'. The main content area shows the article title 'Two-hundred-terabyte maths proof is largest ever' by Evelyn Lamb, dated 26 May 2016. A share icon is visible to the right of the article title.

nature
International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive | Volume 534 | Issue 7605 | News | Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

Beitrag von
M. Heule
O. Kullmann
V. Marek

Kann die Menge $N = \{1, 2, \dots, n\}$ in zwei Untermengen zerlegt werden, so dass keine Untermenge ein pythagoräisches Triple (a, b, c) enthält mit $a^2 + b^2 = c^2$?

$n = 40$

$$3^2 + 4^2 = 5^2$$

$$5^2 + 12^2 = 13^2$$

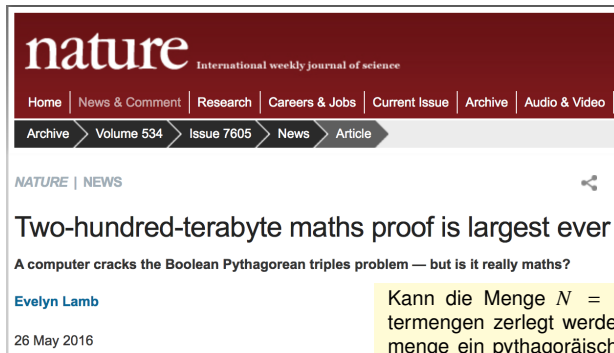
$$8^2 + 15^2 = 17^2$$

$$7^2 + 24^2 = 25^2$$

$$20^2 + 21^2 = 29^2$$

$$12^2 + 35^2 = 37^2$$

(wähle die Farbe der anderen Zahlen beliebig)



The screenshot shows the top navigation bar of the Nature journal website. The header includes the 'nature' logo and the tagline 'International weekly journal of science'. Below this is a row of links: Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. A secondary navigation bar highlights 'Archive', 'Volume 534', 'Issue 7605', 'News', and 'Article'. The main content area shows the article title 'Two-hundred-terabyte maths proof is largest ever' by Evelyn Lamb, dated 26 May 2016. A sub-header reads 'A computer cracks the Boolean Pythagorean triples problem — but is it really maths?'. A share icon is visible to the right of the article title.

nature
International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive | Volume 534 | Issue 7605 | News | Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

Beitrag von
M. Heule
O. Kullmann
V. Marek

Shown by SAT-Solver:

For $n \geq 7825$ consistent bi-coloring becomes impossible.

Kann die Menge $N = \{1, 2, \dots, n\}$ in zwei Untermengen zerlegt werden, so dass keine Untermenge ein pythagoräisches Triple (a, b, c) enthält mit $a^2 + b^2 = c^2$?

$n = 40$

$$3^2 + 4^2 = 5^2$$

$$5^2 + 12^2 = 13^2$$

$$8^2 + 15^2 = 17^2$$

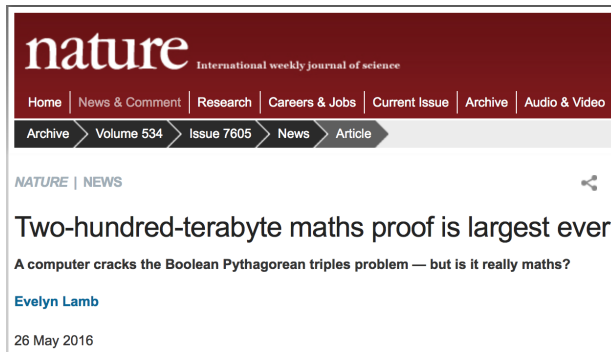
$$7^2 + 24^2 = 25^2$$

$$20^2 + 21^2 = 29^2$$

$$12^2 + 35^2 = 37^2$$

(wähle die Farbe der anderen Zahlen beliebig)

Beispiel einer Erfolgsgeschichte



The screenshot shows the top section of the Nature journal website. The header is dark red with the 'nature' logo in white. Below the logo is the tagline 'International weekly journal of science'. A navigation bar contains links: Home, News & Comment, Research, Careers & Jobs, Current Issue, Archive, and Audio & Video. Below this is a secondary navigation bar with links: Archive, Volume 534, Issue 7605, News (highlighted), and Article. The main content area has a sub-header 'NATURE | NEWS' and a share icon. The article title is 'Two-hundred-terabyte maths proof is largest ever'. Below the title is a subtitle: 'A computer cracks the Boolean Pythagorean triples problem — but is it really maths?'. The author's name 'Evelyn Lamb' is listed in blue. The date '26 May 2016' is at the bottom left.

nature International weekly journal of science

Home | News & Comment | Research | Careers & Jobs | Current Issue | Archive | Audio & Video

Archive | Volume 534 | Issue 7605 | **News** | Article

NATURE | NEWS

Two-hundred-terabyte maths proof is largest ever

A computer cracks the Boolean Pythagorean triples problem — but is it really maths?

Evelyn Lamb

26 May 2016

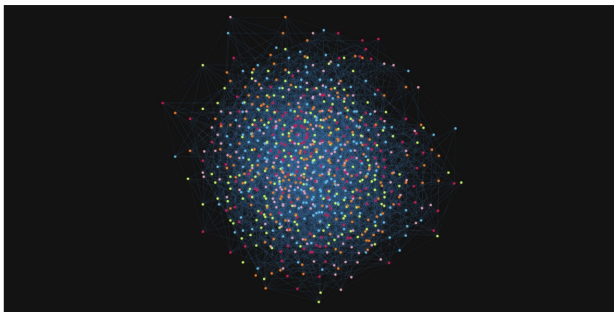
Beitrag von
M. Heule
O. Kullmann
V. Marek

The Science of Brute Force, M.J.H. Heule, O. Kullmann, Communications of the ACM, Vol. 60 No. 8, Pages 70-79, 2017, DOI:10.1145/3107239

<https://cacm.acm.org/magazines/2017/8/219606-the-science-of-brute-force/fulltext>

EVELYN LAMB SCIENCE 04.30.18 09:00 AM

AN ANTI-AGING PUNDIT SOLVES A DECADES-OLD MATH PROBLEM



This 826-vertex graph requires at least five colors to ensure that no two connected vertices are the same shade. (Click [here](#) for a high-resolution version.)

 OLENA SHMAHALO/QUANTA MAGAZINE; SOURCE: [MARIJN HEULE](#)

Übergeordnetes Thema: Erfüllbarkeitsproblem (SAT) der Aussagenlogik

- ▶ **P vs. NP** (1 Million Dollar Frage, Clay Mathematics Institute)
- ▶ **SAT ist NP-vollständig**
- ▶ **SAT-Solver** trotzdem sehr erfolgreich in der Praxis

Fokus der Vorlesung: DPLL-Algorithmus

- ▶ Intelligente Tiefensuche (Backtracking)
- ▶ Literatur
 - ▶ Martin Davis, Hilary Putnam. **A Computing Procedure for Quantification Theory**. J.ACM 7(3): 201-215 (1960)
 - ▶ Martin Davis, George Logemann, Donald W. Loveland. **A Machine Program for Theorem Proving**. Commun. ACM 5(7):394-397 (1962)
 - ▶ ...
 - ▶ Robert Nieuwenhuis, Albert Oliveras, Cesare Tinelli. **Solving SAT and SAT Modulo Theories: From an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T)**. J. ACM 53(6): 937-977 (2006)
 - ▶ ...

Übergeordnetes Thema: Erfüllbarkeitsproblem (SAT) der Aussagenlogik

- ▶ **P vs. NP** (1 Million Dollar Frage, Clay Mathematics Institute)
- ▶ **SAT** ist **NP-vollständig**
- ▶ **SAT-Solver** trotzdem sehr erfolgreich in der Praxis

Fokus der Vorlesung: DPLL-Algorithmus

- ▶ Intelligente Tiefensuche (Backtracking)
- ▶ Literatur
 - ▶ Martin Davis, Hilary Putnam. **A Computing Procedure for Quantification Theory**. J.ACM 7(3): 201-215 (1960)
 - ▶ Martin Davis, George Logemann, Donald W. Loveland. **A Machine Program for Theorem Proving**. Commun. ACM 5(7):394-397 (1962)
 - ▶ ...
 - ▶ Robert Nieuwenhuis, Albert Oliveras, Cesare Tinelli. **Solving SAT and SAT Modulo Theories: From an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T)**. J. ACM 53(6): 937-977 (2006)
 - ▶ ...

Syntax Aussagenlogik

$$s, t ::= \top \mid \perp \mid A \mid \neg s \mid s \vee t \mid s \wedge t \mid s \rightarrow t \mid s \leftrightarrow t$$

Semantik Aussagenlogik: Abbildung nach T (true) oder F (false)

s	t	\top	\perp	$\neg s$	$s \vee t$	$s \wedge t$	$s \rightarrow t$	$s \leftrightarrow t$
T	T	T	F	F	T	T	T	T
T	F	T	F	F	T	F	F	F
F	T	T	F	T	T	F	T	F
F	F	T	F	T	F	F	T	T

Klauselnormalform bzw. Konjunktive Normalform (CNF)

Beispielformel:

$$(A \wedge \neg B) \leftrightarrow (A \vee B)$$

Beispielformel in CNF:

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Syntax Aussagenlogik

$$s, t ::= \top \mid \perp \mid A \mid \neg s \mid s \vee t \mid s \wedge t \mid s \rightarrow t \mid s \leftrightarrow t$$

Semantik Aussagenlogik: Abbildung nach T (true) oder F (false)

s	t	\top	\perp	$\neg s$	$s \vee t$	$s \wedge t$	$s \rightarrow t$	$s \leftrightarrow t$
T	T	T	F	F	T	T	T	T
T	F	T	F	F	T	F	F	F
F	T	T	F	T	T	F	T	F
F	F	T	F	T	F	F	T	T

Klauselnormalform bzw. Konjunktive Normalform (CNF)

Beispielformel:

$$(A \wedge \neg B) \leftrightarrow (A \vee B)$$

Beispielformel in CNF:

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Syntax Aussagenlogik

$$s, t ::= \top \mid \perp \mid A \mid \neg s \mid s \vee t \mid s \wedge t \mid s \rightarrow t \mid s \leftrightarrow t$$

Semantik Aussagenlogik: Abbildung nach T (rue) oder F (alse)

s	t	\top	\perp	$\neg s$	$s \vee t$	$s \wedge t$	$s \rightarrow t$	$s \leftrightarrow t$
T	T	T	F	F	T	T	T	T
T	F	T	F	F	T	F	F	F
F	T	T	F	T	T	F	T	F
F	F	T	F	T	F	F	T	T

Klauselnormalform bzw. Konjunktive Normalform (CNF)

Beispielformel:

$$(A \wedge \neg B) \leftrightarrow (A \vee B)$$

Beispielformel in CNF:

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Syntax Aussagenlogik

$$s, t ::= \top \mid \perp \mid A \mid \neg s \mid s \vee t \mid s \wedge t \mid s \rightarrow t \mid s \leftrightarrow t$$

Semantik Aussagenlogik: Abbildung nach T (rue) oder F (alse)

s	t	\top	\perp	$\neg s$	$s \vee t$	$s \wedge t$	$s \rightarrow t$	$s \leftrightarrow t$
T	T	T	F	F	T	T	T	T
T	F	T	F	F	T	F	F	F
F	T	T	F	T	T	F	T	F
F	F	T	F	T	F	F	T	T

Klauselnormalform bzw. Konjunktive Normalform (CNF)

Beispielformel:

$$(A \wedge \neg B) \leftrightarrow (A \vee B)$$

Beispielformel in CNF:

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Notation: Konjunktion von Klausel, Klauseln, Literale

CNF

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Notation für Klauseln

$$A\bar{B} \wedge \bar{A}\bar{B} \wedge \bar{B}$$

Notation für Klausel-Listen

$$A\bar{B}, \bar{A}\bar{B}, \bar{B}$$

Annahme (für Rest der Vorlesung)

Darstellung von Klauseln und Klausel-Listen: **Kommutativ und Assoziativ**

Belegungen $[\psi]$ am Beispiel:

 $[A\bar{B}]$ repräsentiert die **Belegung** $\{A \longrightarrow \text{T}, B \longrightarrow \text{F}\}$ $[\bar{B}]$ ist eine **partielle Belegung** für $A\bar{B}, \bar{A}\bar{B}, \bar{B}$ $[\psi] \models s$ ist **Notation** für: Belegung $[\psi]$ erfüllt Formel s $[\psi]$ wird dann auch als **Modell für** s bezeichnet

Regelsysteme/Transitionssysteme

Grundkenntnisse werden vorausgesetzt

Notation: Konjunktion von Klausel, Klauseln, Literale

CNF

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Notation für Klauseln

$$A\bar{B} \wedge \bar{A}\bar{B} \wedge \bar{B}$$

Notation für Klausel-Listen

$$A\bar{B}, \bar{A}\bar{B}, \bar{B}$$

Annahme (für Rest der Vorlesung)Darstellung von Klauseln und Klausel-Listen: **Kommutativ und Assoziativ****Belegungen $[\psi]$ am Beispiel:** $[A\bar{B}]$ repräsentiert die **Belegung** $\{A \longrightarrow \text{T}, B \longrightarrow \text{F}\}$ $[\bar{B}]$ ist eine **partielle Belegung** für $A\bar{B}, \bar{A}\bar{B}, \bar{B}$ $[\psi] \models s$ ist **Notation** für: Belegung $[\psi]$ erfüllt Formel s $[\psi]$ wird dann auch als **Modell für s** bezeichnet**Regelsysteme/Transitionssysteme**

Grundkenntnisse werden vorausgesetzt

Notation: Konjunktion von Klausel, Klauseln, Literale

CNF

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Notation für Klauseln

$$A\bar{B} \wedge \bar{A}\bar{B} \wedge \bar{B}$$

Notation für Klausel-Listen

$$A\bar{B}, \bar{A}\bar{B}, \bar{B}$$

Annahme (für Rest der Vorlesung)Darstellung von Klauseln und Klausel-Listen: **Kommutativ und Assoziativ****Belegungen $[\psi]$ am Beispiel:** $[A\bar{B}]$ repräsentiert die **Belegung** $\{A \longrightarrow \text{T}, B \longrightarrow \text{F}\}$ $[\bar{B}]$ ist eine **partielle Belegung** für $A\bar{B}, \bar{A}\bar{B}, \bar{B}$ $[\psi] \models s$ ist **Notation** für: Belegung $[\psi]$ erfüllt Formel s $[\psi]$ wird dann auch als **Modell für s** bezeichnet**Regelsysteme/Transitionssysteme**

Grundkenntnisse werden vorausgesetzt

Notation: Konjunktion von Klausel, Klauseln, Literale

CNF

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Notation für Klauseln

$$A\bar{B} \wedge \bar{A}\bar{B} \wedge \bar{B}$$

Notation für Klausel-Listen

$$A\bar{B}, \bar{A}\bar{B}, \bar{B}$$

Annahme (für Rest der Vorlesung)

Darstellung von Klauseln und Klausel-Listen: **Kommutativ und Assoziativ**

Belegungen $[\psi]$ am Beispiel:

 $[A\bar{B}]$ repräsentiert die **Belegung** $\{A \longrightarrow \text{T}, B \longrightarrow \text{F}\}$ $[\bar{B}]$ ist eine **partielle Belegung** für $A\bar{B}, \bar{A}\bar{B}, \bar{B}$ $[\psi] \models s$ ist **Notation** für: Belegung $[\psi]$ erfüllt Formel s $[\psi]$ wird dann auch als **Modell** für s bezeichnet

Regelsysteme/Transitionssysteme

Grundkenntnisse werden vorausgesetzt

Notation: Konjunktion von Klausel, Klauseln, Literale

CNF

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Notation für Klauseln

$$A\bar{B} \wedge \bar{A}\bar{B} \wedge \bar{B}$$

Notation für Klausel-Listen

$$A\bar{B}, \bar{A}\bar{B}, \bar{B}$$

Annahme (für Rest der Vorlesung)

Darstellung von Klauseln und Klausel-Listen: **Kommutativ und Assoziativ**

Belegungen $[\psi]$ am Beispiel:

 $[A\bar{B}]$ repräsentiert die **Belegung** $\{A \longrightarrow \text{T}, B \longrightarrow \text{F}\}$ $[\bar{B}]$ ist eine **partielle Belegung** für $A\bar{B}, \bar{A}\bar{B}, \bar{B}$ $[\psi] \models s$ ist **Notation** für: Belegung $[\psi]$ erfüllt Formel s $[\psi]$ wird dann auch als **Modell für** s bezeichnet

Regelsysteme/Transitionssysteme

Grundkenntnisse werden vorausgesetzt

Notation: Konjunktion von Klausel, Klauseln, Literale

CNF

$$(A \vee \neg B) \wedge (\neg A \vee \neg B) \wedge \neg B$$

Notation für Klauseln

$$A\bar{B} \wedge \bar{A}\bar{B} \wedge \bar{B}$$

Notation für Klausel-Listen

$$A\bar{B}, \bar{A}\bar{B}, \bar{B}$$

Annahme (für Rest der Vorlesung)

Darstellung von Klauseln und Klausel-Listen: **Kommutativ und Assoziativ**

Belegungen $[\psi]$ am Beispiel:

 $[A\bar{B}]$ repräsentiert die **Belegung** $\{A \longrightarrow \text{T}, B \longrightarrow \text{F}\}$ $[\bar{B}]$ ist eine **partielle Belegung** für $A\bar{B}, \bar{A}\bar{B}, \bar{B}$ $[\psi] \models s$ ist **Notation** für: Belegung $[\psi]$ erfüllt Formel s $[\psi]$ wird dann auch als **Modell für** s bezeichnet

Regelsysteme/Transitionssysteme

Grundkenntnisse werden vorausgesetzt

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
\square	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$$[\bar{B}C] \quad \underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark} \quad \text{—alle Klauseln sind erfüllt—}$$

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$$[\psi] \quad \phi \quad (\text{Solved})$$

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
\square	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$$[\bar{B}C] \quad \underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark} \quad \text{—alle Klauseln sind erfüllt—}$$

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$$[\psi] \quad \phi \quad (\text{Solved})$$

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C]$ $\underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark}$ —alle Klauseln sind erfüllt—

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi \quad (\text{Solved})$

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C]$ $\underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark}$ —alle Klauseln sind erfüllt—

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi$ (Solved)

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C]$ $\underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark}$ —alle Klauseln sind erfüllt—

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi$ (Solved)

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C] \quad \underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark} \quad \text{—alle Klauseln sind erfüllt—}$

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi \quad (\text{Solved})$

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C]$ $\underbrace{A\bar{B}C}$, $\underbrace{\bar{B}}$, $\underbrace{C\bar{D}}$, \underbrace{BCD} —alle Klauseln sind erfüllt—
 ✓ ✓ ✓ ✓

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi$ (Solved)

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C] \quad \underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark} \quad \text{—alle Klauseln sind erfüllt—}$

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi \quad (\text{Solved})$

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C]$ $\underbrace{A\bar{B}C}$, $\underbrace{\bar{B}}$, $\underbrace{C\bar{D}}$, \underbrace{BCD} —alle Klauseln sind erfüllt—

✓ ✓ ✓ ✓

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi$ (Solved)

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C] \quad \underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark} \quad \text{—alle Klauseln sind erfüllt—}$

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi \quad (\text{Solved})$

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C]$ $\underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark}$ —alle Klauseln sind erfüllt—

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi$ (Solved)

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C]$ $\underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark}$ —alle Klauseln sind erfüllt—

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi \quad (\text{Solved})$

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabeformel	(CNF & Initialisierung)
[]	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Schritt 1)
	...	(Schritt 2)
	...	
	...	(Schritt n)
$[\bar{B}C]$	$A\bar{B}C, \bar{B}, C\bar{D}, BCD$	(Solved)

Kriterium für “Solved”?

$[\bar{B}C]$ $\underbrace{A\bar{B}C}_{\checkmark}, \underbrace{\bar{B}}_{\checkmark}, \underbrace{C\bar{D}}_{\checkmark}, \underbrace{BCD}_{\checkmark}$ —alle Klauseln sind erfüllt—

(es gilt dann: $[\bar{B}C] \models \text{Eingabeformel}$)

Solved-Regel

$[\psi] \quad \phi \quad (\text{Solved})$

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	(CNF & Initialisierung)
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	(CNF & Initialisierung)
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	(CNF & Initialisierung)
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	(CNF & Initialisierung)
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

Unit-Propagation-Regel:

$[\psi] \quad \phi, \varphi L \quad (\text{Unit-Propagation})$

Bedingung:

\downarrow

$[\psi] \models \neg\varphi$

$L \notin [\psi]$

$[\psi L] \quad \phi, \varphi L$

Verarbeite neue Belegungsinformation

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	(CNF & Initialisierung)
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	(CNF & Initialisierung)
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	(CNF & Initialisierung)
$[\]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

Pure-Literal-Regel

$[\psi] \quad \phi, \varphi L \quad (\text{Pure-Literal})$

Bedingung:

↓

L nur mit einheitlicher Polarität in $\phi, \varphi L$

L undefiniert in $[\psi]$

$[\psi L] \quad \phi, \varphi L$

Verarbeite neue Belegungsinformation

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	(CNF & Initialisierung)
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, \cancel{C}\bar{G}F, \bar{C}\bar{F}G, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}}FG, \cancel{\bar{C}}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}}\bar{F}G, \bar{D}E, \cancel{\bar{F}}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}}\bar{F}G, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, \cancel{C}\bar{G}F, \bar{C}\bar{F}G, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}}FG, \cancel{\bar{C}}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}}\bar{F}G, \bar{D}E, \cancel{\bar{F}}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}}\bar{F}G, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, \cancel{C\bar{G}F}, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

Split-Regel:

$[\psi] \quad \phi \quad (\text{Split})$

Bedingung:

\downarrow

L und \bar{L} kommen in ϕ vor

L undefiniert in $[\psi]$

$[\psi L_b] \quad \phi$

Verarbeite neue Belegungsinformation

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{F}G, \cancel{\bar{F}}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}}\bar{F}G, \bar{D}E, \cancel{\bar{F}}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}}\bar{F}G, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{F}G, \cancel{\bar{F}}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{F}}G, \bar{D}E, \cancel{\bar{F}}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{F}}G, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$\{\bar{B}\bar{D}C_bF_b\}$	$A\bar{B}, \bar{B}, \cancel{BCF}, \cancel{C\bar{G}F}, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$\{\bar{B}\bar{D}C_bF_bG\}$	$A\bar{B}, \bar{B}, \cancel{BCF}, \cancel{C\bar{G}F}, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

Backtrack Rule:

$[\psi L_b \psi'] \quad \phi, \varphi \quad (\text{Backtrack})$

Bedingung:

\downarrow

$[\psi L_b \psi'] \models \neg \varphi$

ψ' enthält keine mit “b” markierten Literale

$[\psi \bar{L}] \quad \phi, \varphi$

Verarbeite neue Belegungsinformation

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	(CNF & Initialisierung)
$[\]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Solved)

Solved-Regel

$[\psi] \quad \phi \quad (\text{Solved})$

Bedingung: $[\psi] \models \phi$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

Fail-Regel:

$[\psi] \quad \phi, \varphi \quad (\text{Split})$

Bedingung:

\downarrow

$[\psi] \models \neg\varphi$

ψ enthält keine Backtrack-Literale

$[] \quad \text{Fail}$

DPLL-Algorithmus — Entwicklung am Beispiel

	Eingabe-Formel	
$[]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(CNF & Initialisierung)
$[\bar{B}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
$[\bar{B}\bar{D}]$	$A\bar{B}, \bar{B}, \cancel{BCF}, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Pure-Literal)
$[\bar{B}\bar{D}C_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}FG}, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \bar{F}\bar{G}$	(Split)
$[\bar{B}\bar{D}C_bF_b]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Split)
$[\bar{B}\bar{D}C_bF_bG]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \cancel{\bar{C}\bar{F}G}, \bar{D}E, \cancel{\bar{F}\bar{G}}$	(Unit-Propagation)
$[\bar{B}\bar{D}C_b\bar{F}]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \cancel{\bar{C}\bar{F}G}, \bar{C}FG, \bar{D}E, \bar{F}\bar{G}$	(Backtrack)
$[\bar{B}\bar{D}C_b\bar{F}G]$	$A\bar{B}, \bar{B}, BCF, C\bar{G}F, \bar{C}FG, \bar{C}\bar{F}G, \bar{D}E, \bar{F}\bar{G}$	(Unit-Propagation)
		(Solved)

Abstrakter DPLL Algorithmus (Regelbasierte Darstellung)

Wende die Regeln erschöpfend an mit folgender Priorität:

1. Solved
2. Fail
3. Backtrack
4. Unit-Propagation
5. Pure-Literal
6. Split

DPLL Algorithmus am Beispiel — Kurzes Beispiel, alle Regeln

[]	an , ãn , ãñ , bn	(Pure Literal)
[b]	an , ãn , ãñ , bn	(Split)
[ba]	an , ãn , ãñ , bn	(Unit Propagation)
[ban]	an , ãn , ãñ , bn	(Widerspruch: n&ñ, Backtracking)
[bã]	an , ãn , ãñ , bn	(Unit Propagation)
[bãn]	an , ãn , ãñ , bn	(Solved)

Möglichkeiten zur weiteren Verbesserung bzw. Optimierung

- ▶ Heuristiken für Split: z.B. wähle das am häufigsten vorkommende Literal (Vorheriges Beispiel: wähle F zuerst – diese Wahl verhindert Backtracking und generiert eine alternative Belegung (Übungsaufgabe))
- ▶ Nicht-chronologisches Backtracking
- ▶ Klausel-Lernen beim Backtracking
- ▶ Effiziente Datenstrukturen
- ▶ Indexing-Techniken, Watchlist
- ▶ ...

Korrektheit und Vollständigkeit: für abstraktes System in Vorlesung

Systeme (z.B. zChaff, MiniSat, PicoSat, Limmat, Lingeling)

Anwendungsbeispiele

Satisfiability modulo Theories