Einführung in die Technische Informatik

SAT Beweiser

Robert Wille

Motivation

- Bisher:
  - Formale Methoden zum Nachweis der Korrektheit
  - Vielversprechende Technik:
    Löser für Boolesche Erfüllbarkeit
  - Betrachtet als „Black Box"
- Jetzt:
  - Einführung in die Kerntechnologien
  - Erweiterungen des Erfüllbarkeitsproblems

SAT-Algorithmus

$(x1 \vee x2)$

$(x3 \vee x4)$

$(x1 \vee x3 \vee \neg x4)$

➔ bis zu 2^n Möglichkeiten

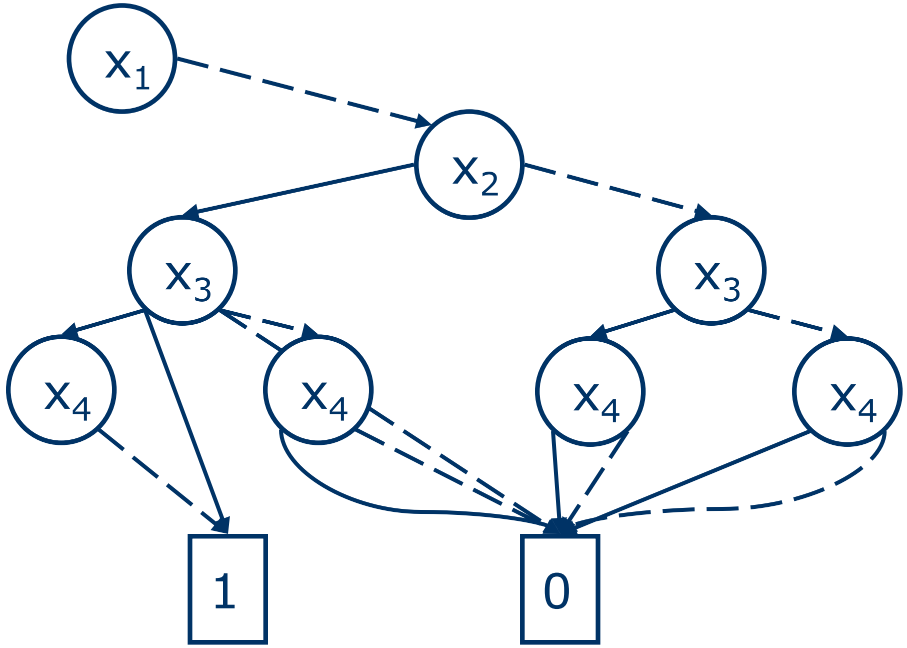
1	2	3	4	*f*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1

SAT-Algorithmus (DPLL)

$$(x1 \vee x2)$$

$$(x3 \vee x4)$$

$$(x1 \vee x3 \vee \neg x4)$$




```
while decide() do
  propagation()
  if (conflict()) then
    if conflict_analysis() then
      backtrack()
    else
      return UNSAT
done
return SAT;
```

SAT-Algorithmus (DPLL)

$\omega1 = (\neg x1 \vee x2)$
$\omega2 = (\neg x1 \vee x3 \vee x9)$
$\omega3 = (\neg x2 \vee \neg x3 \vee x4)$
$\omega4 = (\neg x4 \vee x5 \vee x10)$
$\omega5 = (\neg x4 \vee x6 \vee x11)$
$\omega6 = (\neg x5 \vee \neg x6)$
$\omega7 = (x1 \vee x7 \vee \neg x12)$
$\omega8 = (x1 \vee x8)$
$\omega9 = (\neg x7 \vee \neg x8 \vee \neg x13)$

```
while decide() do
   propagation()
   if (conflict()) then
      if conflict_analysis() then
         backtrack()
      else
         return UNSAT
done
return SAT;
```

Entscheidungsheuristik

$\omega 1 = (\neg x1 \vee x2)$
$\omega 2 = (\neg x1 \vee x3 \vee x9)$
$\omega 3 = (\neg x2 \vee \neg x3 \vee x4)$
$\omega 4 = (\neg x4 \vee x5 \vee x10)$
$\omega 5 = (\neg x4 \vee x6 \vee x11)$
$\omega 6 = (\neg x5 \vee \neg x6)$
$\omega 7 = (x1 \vee x7 \vee \neg x12)$
$\omega 8 = (x1 \vee x8)$
$\omega 9 = (\neg x7 \vee \neg x8 \vee \neg x13)$

x1:	4	x2:	2
x3:	2	x4:	2
x5:	2	x6:	2
x7:	1	x8:	2
x9:	1	x10:	1
x11:	1	x12:	1
x13:	1		

➔ nach Häufigkeit
➔ nach Vorkommen der Phasen
➔ …

Propagieren

- $\omega1 = (\boxed{\neg x1} \vee \boxed{x2})$
- $\omega2 = (\neg x1 \vee \boxed{x3} \vee \boxed{x9})$
- $\omega3 = (\boxed{\neg x2} \vee \boxed{\neg x3} \vee x4)$
- $\omega4 = (\neg x4 \vee \boxed{x5} \vee \boxed{x10})$
- $\omega5 = (\neg x4 \vee \boxed{x6} \vee \boxed{x11})$
- $\omega6 = (\boxed{\neg x5} \vee \boxed{\neg x6})$
- $\omega7 = (\boxed{x1} \vee \boxed{x7} \vee \neg x12)$
- $\omega8 = (\boxed{x1} \vee \boxed{x8})$
- $\omega9 = (\neg x7 \vee \boxed{\neg x8} \vee \boxed{\neg x13})$

x1=1 soll propagiert werden

➔ alle Klauseln betrachten

➔ alle Klauseln betrachten, in denen Variable x1 vorkommt

➔ alle Klauseln betrachten, in denen ¬x1 vorkommt

➔ Two Watch Literal Scheme

Konfliktanalyse

$\omega 1 = (\neg x1 \vee x2)$

$\omega 2 = (\neg x1 \vee x3 \vee x9)$

$\omega 3 = (\neg x2 \vee \neg x3 \vee x4)$

$\omega 4 = (\neg x4 \vee x5 \vee x10)$

$\omega 5 = (\neg x4 \vee x6 \vee x11)$

$\omega 6 = (\neg x5 \vee \neg x6)$

$\omega 7 = (x1 \vee x7 \vee \neg x12)$

$\omega 8 = (x1 \vee x8)$

$\omega 9 = (\neg x7 \vee \neg x8 \vee \neg x13)$

```
while decide() do
   propagation()
   if (conflict()) then
      if conflict_analysis() then
         backtrack()
      else
         return UNSAT
done
return SAT;
```

Konfliktanalyse

$\omega1 = \cancel{(\neg x1 \vee x2)}$

$\omega2 = \cancel{(\neg x1 \vee x3 \vee x9)}$

$\omega3 = \cancel{(\neg x2 \vee \neg x3 \vee x4)}$

$\omega4 = \cancel{(\neg x4 \vee x5 \vee x10)}$

$\omega5 = \cancel{(\neg x4 \vee x6 \vee x11)}$

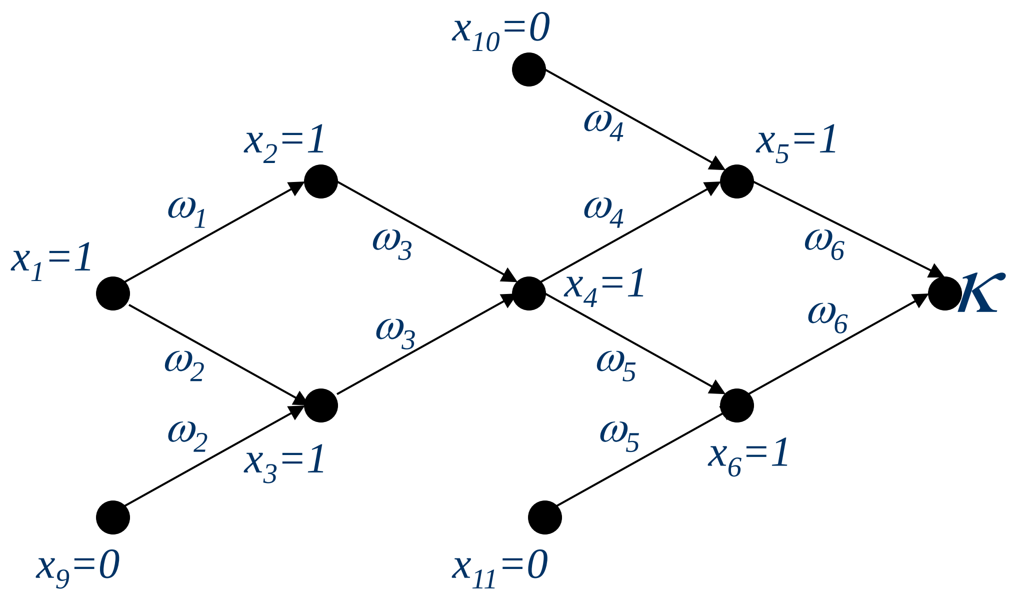
$\omega6 = (\neg x5 \vee \neg x6)$

$\omega7 = \cancel{(x1 \vee x7 \vee \neg x12)}$

$\omega8 = \cancel{(x1 \vee x8)}$

$\omega9 = (\neg x7 \vee \neg x8 \vee \neg\, x13)$

$\omega\kappa = (\neg x1 \vee\ x9 \vee\ x10 \vee\ x11)$



➔ Implikationsgraph
➔ ermöglicht **Lernen** einer Konfliktklausel

Konfliktanalyse

$\omega 1 = (\neg x1 \vee x2)$

$\omega 2 = (\neg x1 \vee x3 \vee \cancel{x9})$

$\omega 3 = (\neg x2 \vee \neg x3 \vee x4)$

$\omega 4 = (\neg x4 \vee x5 \vee \cancel{x10})$

$\omega 5 = (\neg x4 \vee x6 \vee \cancel{x11})$

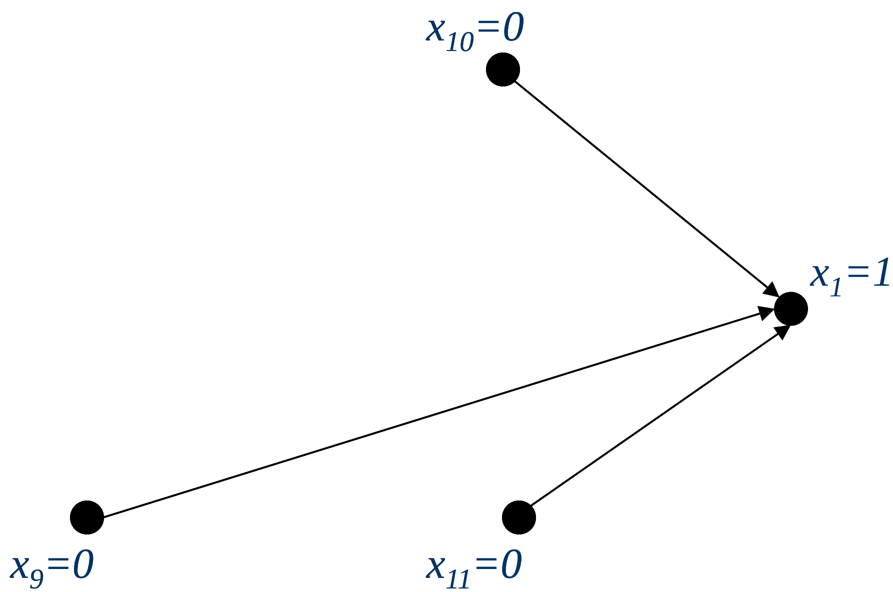
$\omega 6 = (\neg x5 \vee \neg x6)$

$\omega 7 = (x1 \vee x7 \vee \neg x12)$

$\omega 8 = (x1 \vee x8)$

$\omega 9 = (\neg x7 \vee \neg x8 \vee \neg x13)$

$\omega\kappa = (\neg x1 \vee \cancel{x9} \vee \cancel{x10} \vee \cancel{x11})$



➔ Implikationsgraph
➔ ermöglicht **Lernen** einer Konfliktklausel

Konfliktanalyse

$\omega 1 = \cancel{(\neg x1 \vee x2)}$

$\omega 2 = \cancel{(\neg x1 \vee x3 \vee x9)}$

$\omega 3 = \cancel{(\neg x2 \vee \neg x3 \vee x4)}$

$\omega 4 = \cancel{(\neg x4 \vee x5 \vee x10)}$

$\omega 5 = \cancel{(\neg x4 \vee x6 \vee x11)}$

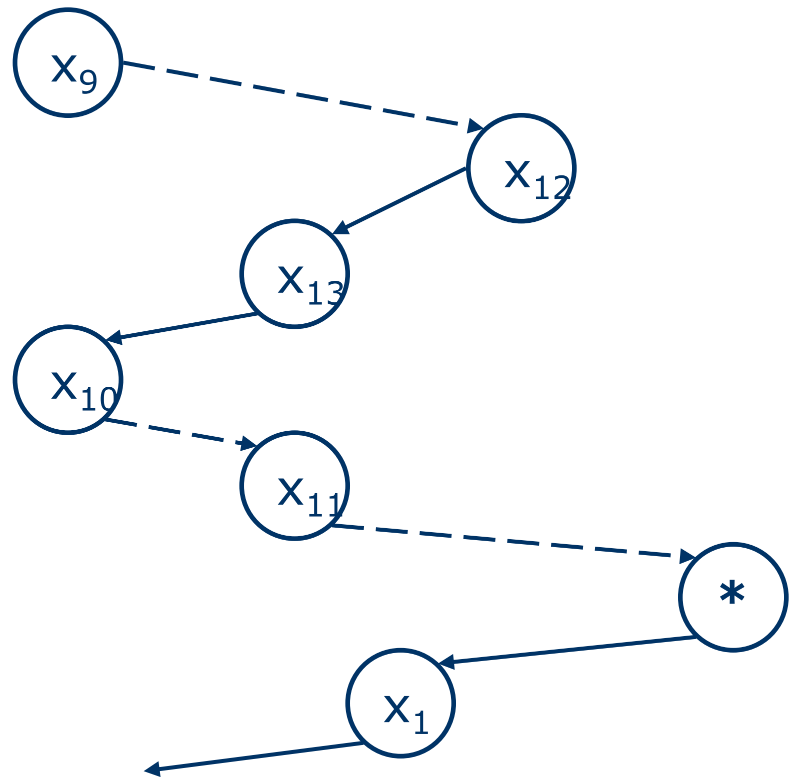
$\omega 6 = (\neg x5 \vee \neg x6)$

$\omega 7 = \cancel{(x1 \vee x7 \vee \neg x12)}$

$\omega 8 = \cancel{(x1 \vee x8)}$

$\omega 9 = (\neg x7 \vee \neg x8 \vee \neg x13)$

…



x2=1 x3=1

x4=1 x5=1

x6=1

κ

Konfliktanalyse

$\omega1 = \cancel{(\neg x1 \vee x2)}$

$\omega2 = \cancel{(\neg x1 \vee x3 \vee \cancel{x9})}$

$\omega3 = (\neg x2 \vee \neg x3 \vee x4)$

$\omega4 = (\neg x4 \vee x5 \vee \cancel{x10})$

$\omega5 = (\neg x4 \vee x6 \vee \cancel{x11})$

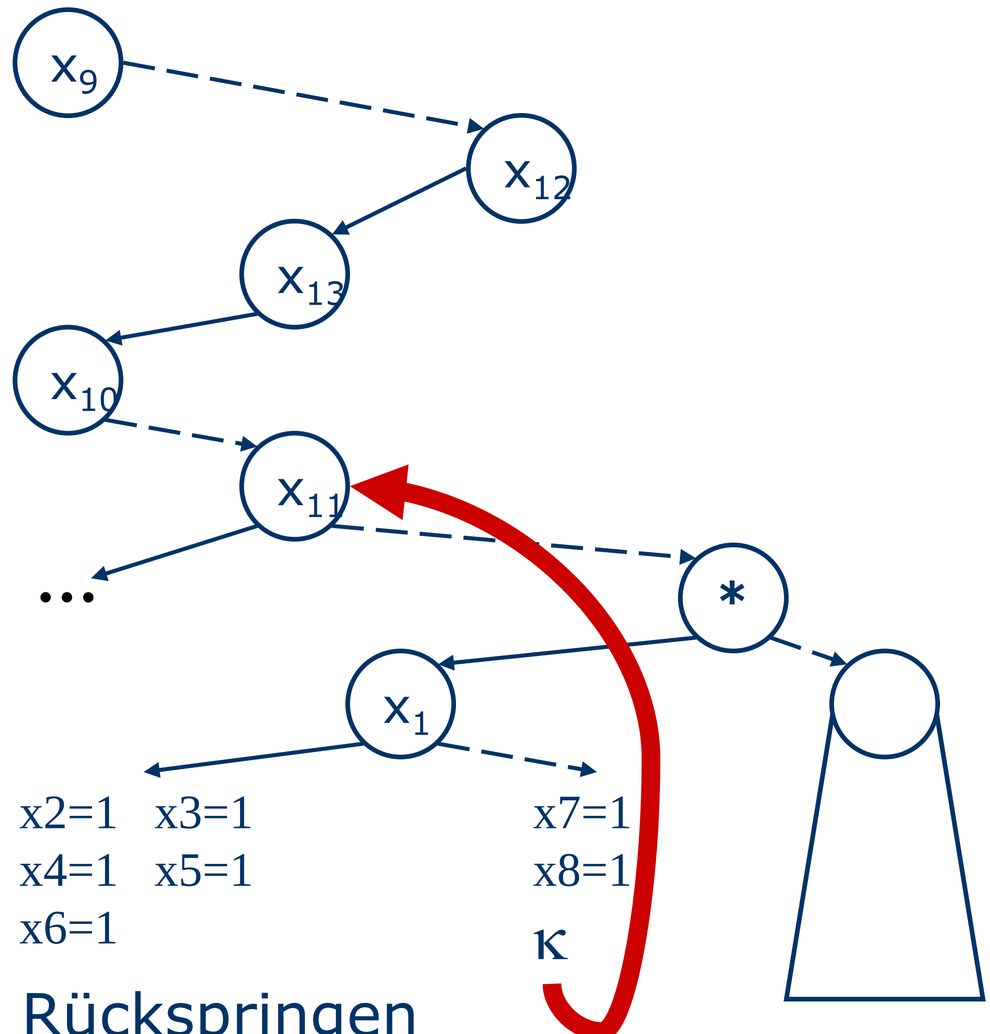
$\omega6 = (\neg x5 \vee \neg x6)$

$\omega7 = \cancel{(\cancel{x1} \vee x7 \vee \cancel{\neg x12})}$

$\omega8 = \cancel{(\cancel{x1} \vee x8)}$

$\omega9 = (\neg \cancel{x7} \vee \neg \cancel{x8} \vee \neg \cancel{x13})$

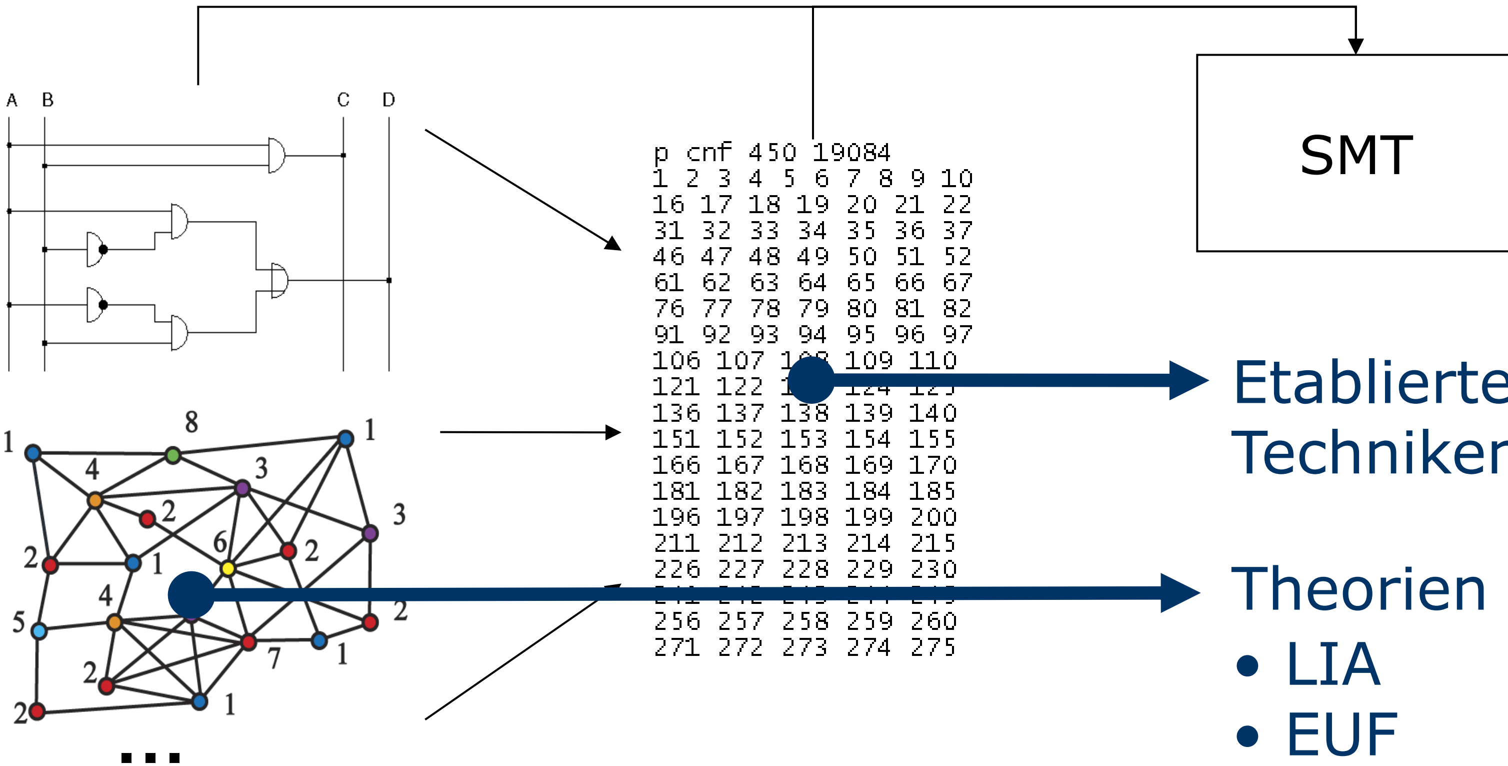
...



➔ nicht-chronologisches Rückspringen

SMT Solver #1

SMT Solver #1



Etablierte Techniken

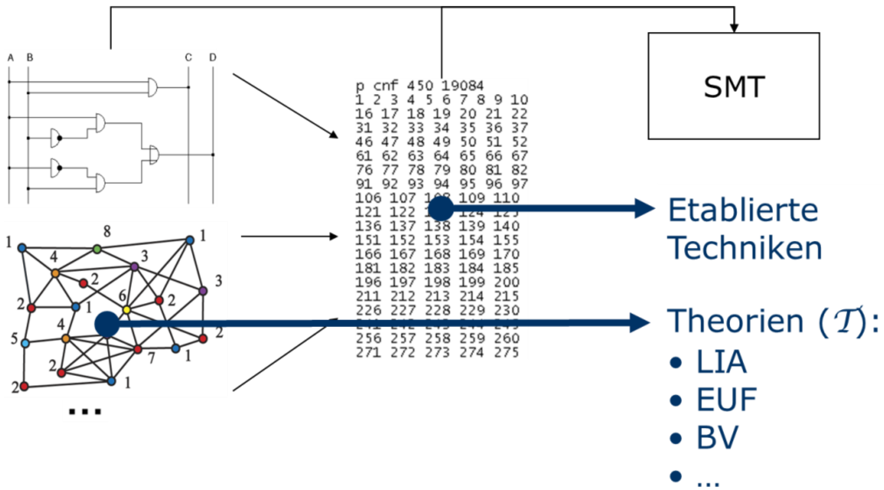
Theorien ($\mathcal{T}$):
- LIA
- EUF
- BV
- …

➔ Erweiterung durch Theorien

SMT Solver #2

- Entscheidung der Erfüllbarkeit unter Berücksichtigung einer darüberliegenden Theorie (SAT **Modulo** Theory)

- Besteht aus:



→ SAT solver (Enumerator)

→ Theory solver ($\mathcal{T}$-solver)

SMT-Algorithm

- Illustriert am Beispiel der Theorie „Equality with Uninterpr. Fcts.“ (EUF)

$$g(a) = c \wedge (f(g(a)) \neq f(c) \vee g(a) = d) \wedge c \neq d$$

SMT-Algorithm

- Illustriert am Beispiel der Theorie „Equality with Uninterpr. Fcts.“ (EUF)

$$\underbrace{g(a) = c}_{1} \wedge (\underbrace{f(g(a)) \neq f(c)}_{\overline{2}} \vee \underbrace{g(a) = d}_{3}) \wedge \underbrace{c \neq d}_{\overline{4}}$$

```
while decide() do
  propagation()

  if (conflict()          ) then
    if conflict_analysis() then
      backtrack()
    else
      return UNSAT
done
return SAT;
```

SMT-Algorithm

- Illustriert am Beispiel der Theorie „Equality with Uninterpr. Fcts.“ (EUF)

$$\underbrace{g(a)=c}_{1} \wedge (\underbrace{f(g(a)) \neq f(c)}_{\overline{2}} \vee \underbrace{g(a)=d}_{3}) \wedge \underbrace{c \neq d}_{\overline{4}}$$

SAT-solver

→ propagate: μ={1,$\overline{4}$}

→ propagate: μ={1, 2, 3, $\overline{4}$}

→ UNSAT

$\mathcal{T}$-solver

→ propagate: μ={1, 2, $\overline{4}$}

→ conflict η= {$\overline{1}$, $\overline{2}$, $\overline{3}$, 4}

Weitere Optimierungen

- Theory-Driven Learning
- Theory-Driven Deduction
- Static Learning
- Exploiting pure T-atoms
- Clause Discharge
- Control on Split Literals
- EQ-Layering
- Weakened Early Pruning

```
do
  while decide() do
    propagation()
    η=call𝒯-Solver(μ)
    if (conflict() || η≠∅) then
      if conflict_analysis() then
        backtrack()
      else
        return UNSAT
  done
return SAT;
```