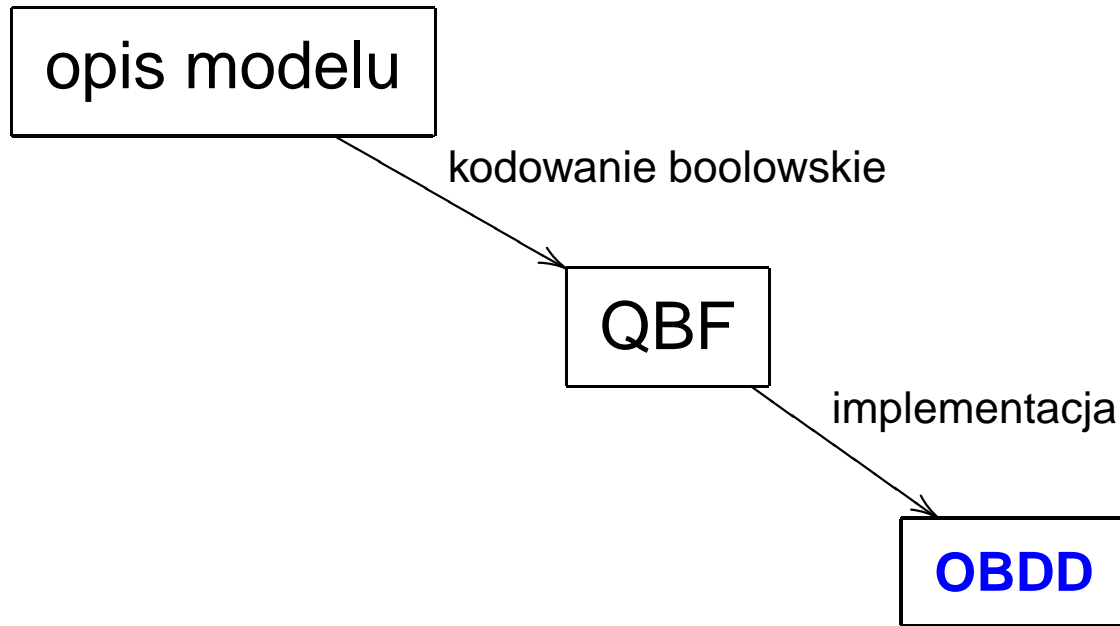


Praktyczne metody weryfikacji

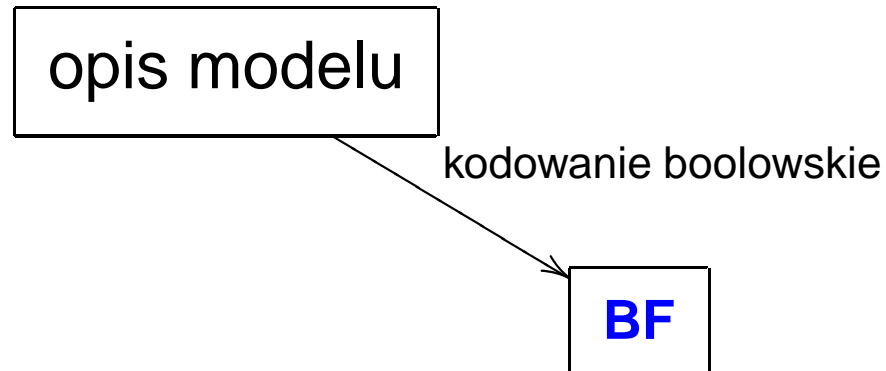
Wykład 9: Weryfikacja ograniczona

Symboliczna weryfikacja modelowa (SMC)



weryfikacja modelowa = operacje na OBDDs

Ograniczona weryfikacja modelowa (BMC)



weryfikacja modelowa = spełnialność formuły boolowskiej

Przykład: $EF \neg p$ (bezpieczeństwo)

pre-SMC:

$$\dots \supseteq \neg p \cup \mathbf{EX}(\neg p \cup \mathbf{EX} \neg p) \supseteq \neg p \cup \mathbf{EX} \neg p \supseteq \neg p \supseteq \text{false}$$

$$\dots \supseteq \neg p \cup \text{pre}(\neg p \cup \text{pre}(\neg p)) \supseteq \neg p \cup \text{pre}(\neg p) \supseteq \neg p \supseteq \text{false}$$

post-SMC:

$$S_0 \subseteq S_0 \cup \text{post}(S_0) \subseteq S_0 \cup \text{post}(S_0 \cup \text{post}(S_0)) \subseteq \dots$$

Ograniczona weryfikacja modelowa

- poszukiwanie kontrprzykładów
ograniczonej długości
- metoda symboliczna
- wykorzystanie **SAT-rozwiązywaczy**
(ang. *SAT-solvers*)

I. Weryfikacja ograniczona

$$M \models \mathbf{E} \phi \quad (\phi \in \mathbf{LTL}^+)$$

np. zamiast $M \models \mathbf{AG} \phi$, sprawdzamy $M \models \mathbf{EF} \neg \phi$

M opisany przez formuły boolowskie:

- $\widehat{R}(x_1, \dots, x_m, x'_1, \dots, x'_m)$
- $\widehat{L}_p(x_1, \dots, x_m), \widehat{S}_0(x_1, \dots, x_m)$

Pomysł: horyzont $k \geq 0$

k kroków systemu

$\Pi \vDash_k \phi$ k kroków wystarcza by stwierdzić, że $\Pi \vDash \phi$

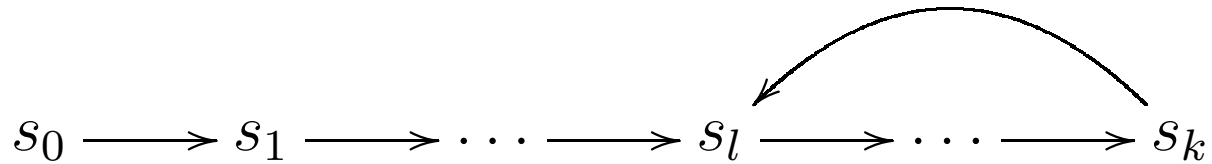
Lem.: $\Pi \vDash_k \phi \implies \Pi \vDash \phi$

Lem.: $M \vDash \mathbf{E} \phi \implies \exists k \geq 0. M \vDash_k \mathbf{E} \phi$

Tw.: $M \vDash \mathbf{E} \phi \iff \exists k \geq 0. M \vDash_k \mathbf{E} \phi$

$$\Pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$$

(k, l) - pęta:



$$s_{k+1+i} = s_{l+i}$$

(k) - pęta)

brak pętli:



Semantyka ograniczona dla k -pętli

$$\Pi \vDash_k \phi \iff \Pi \vDash \phi$$

$$\Pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$$

Semantyka ograniczona dla nie-pętli

$$\Pi \models_k \phi \iff \Pi \models_k^0 \phi$$

$$\Pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \dots$$

$$\models_k^i, \quad 0 \leq i \leq k$$

$$\Pi \models_k^i p \iff p \in L(s_i)$$

$$\Pi \models_k^i \neg p, \quad \phi_1 \wedge \phi_2, \quad \phi_1 \vee \phi_2 \iff \dots$$

$$\Pi \models_k^i \mathbf{X} \phi \iff i < k \text{ i } \Pi \models_k^{i+1} \phi$$

$$\Pi \models_k^i \mathbf{F} \phi \iff \exists j, i \leq j \leq k. \Pi \models_k^j \phi$$

$$\Pi \models_k^i \mathbf{G} \phi \quad \text{nigdy}$$

$$\Pi \models_k^i \phi \mathbf{U} \psi \iff \exists j, i \leq j \leq k. \Pi \models_k^j \psi \wedge \forall l, i \leq l < j. \Pi \models_k^l \phi$$

$$\Pi \models_k^i \phi \mathbf{R} \psi \iff ?$$

$$\phi \mathbf{R} \psi \equiv \neg(\neg\phi \mathbf{U} \neg\psi)$$

$$\Pi \models \phi \mathbf{R} \psi \text{ wtw gdy } \forall i < |\Pi|. (\forall j < i. \Pi^j \models \neg\phi) \implies \Pi^i \models \psi$$

$$\phi \mathbf{R} \psi \equiv \psi \mathbf{U} (\psi \wedge \phi) \vee \mathbf{G} \psi$$

$$\Pi \models_k^i \phi \mathbf{U} \psi \iff \exists j, i \leq j \leq k. \Pi \models_k^j \psi \wedge \forall l, i \leq l < j. \Pi \models_k^l \phi$$

$$\Pi \models_k^i \phi \mathbf{R} \psi \iff \exists j, i \leq j \leq k. \Pi \models_k^j \phi \wedge \forall l, i \leq l \leq j. \Pi \models_k^l \psi$$

Lem.: $\Pi \vDash_k \phi \implies \Pi \vDash \phi$

Lem.: $M \vDash \mathbf{E} \phi \implies \exists k \geq 0. M \vDash_k \mathbf{E} \phi$

Tw.: $M \vDash \mathbf{E} \phi \iff \exists k \geq 0. M \vDash_k \mathbf{E} \phi$

- **jakie k jest wystarczająco duże? średnica grafu?**
- **brak pełności !**
- **efektywność SMC zależy od średnicy grafu !**

$$M \models_k \mathbf{E} \phi \quad (\phi \in \mathbf{LTL}^+)$$

$$M, \phi, k \mapsto [M, \phi]_k$$

$$M \models_k \phi \iff [M, \phi]_k \text{ spełnialna}$$

M opisany przez formuły boolowskie:

- $R(z, z')$
- $L_p(z), S_0(z)$

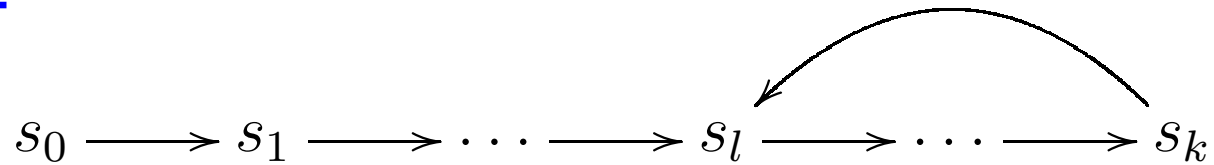
Ścieżka symboliczna „ $z_0 \rightarrow z_1 \rightarrow \dots \rightarrow z_k$ ”

$$[M]_k := S_0(z_0) \wedge \bigwedge_{i=0}^{k-1} R(z_i, z_{i+1})$$

$(k + 1) \cdot m$ zmiennych boolowskich

rozmiar formuły boolowskiej $[M]_k$ jest $\mathcal{O}(k \cdot |M|)$

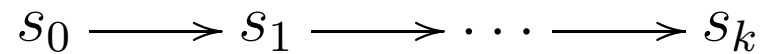
(k, l) - pęta:



$${}_l L_k \equiv R(z_k, z_l)$$

$$L_k \equiv \bigvee_{l=0}^k {}_l L_k$$

brak pętli:



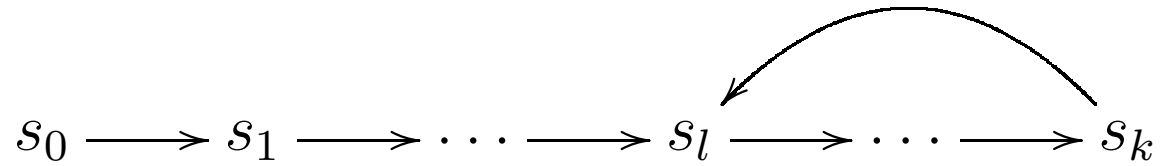
$$[M, \phi]_k := [M]_k \wedge \left((\neg L_k \wedge [\phi]_k^0) \vee \bigvee_{l=0}^k ({}_l L_k \wedge {}_l [\phi]_k^0) \right)$$

${}_l [\phi]_k^0$ – znaczenie ϕ na (k, l) -pętli

$[\phi]_k^0$ – znaczenie ϕ na nie-pętli

rozmiar formuły $[M, \phi]_k$ jest $\mathcal{O}(k \cdot (|M| + |\phi|))$

$$l[\phi]_k^i \quad 0 \leq l, i \leq k$$



$$l[p]_k^i := L_p(z_i)$$

$$l[\neg p]_k^i \quad l[\phi \wedge \psi]_k^i \quad l[\phi \vee \psi]_k^i := \dots$$

$$l[X\phi]_k^i := l[\phi]_k^{\text{succ}(i)}$$

$$\text{succ}(i) = \begin{cases} i + 1 & i < k \\ l & i = k \end{cases}$$

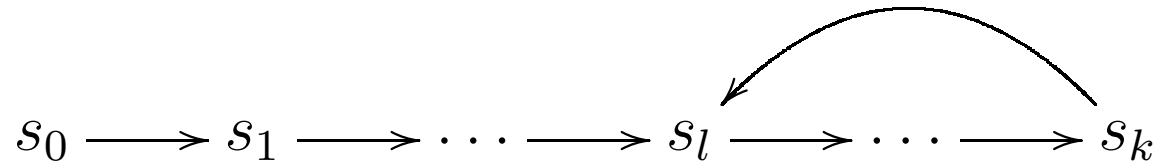
$$\mathbf{F}\phi \equiv \phi \vee \mathbf{X}\mathbf{F}\phi$$

$$\phi \mathbf{U} \psi \equiv \psi \vee (\phi \wedge \mathbf{X}\phi \mathbf{U} \psi)$$

$$\mathbf{G}\phi \equiv \phi \wedge \mathbf{X}\mathbf{G}\phi$$

$$\phi \mathbf{R} \psi \equiv \psi \wedge (\phi \vee \mathbf{X}\phi \mathbf{R} \psi)$$

Przykład:



$${}_l[\mathbf{F} p]_k^0 = L_p(z_0) \vee \dots \vee L_p(z_k)$$

$${}_l[\mathbf{G} p]_k^0 = L_p(z_0) \wedge \dots \wedge L_p(z_k)$$

$${}_l[p \mathbf{U} q]_k^0 = L_q(z_0) \vee (L_p(z_0) \wedge (L_q(z_1) \vee (\dots \\ L_q(z_{k-1}) \vee (L_p(z_{k-1}) \wedge L_q(z_k)) \dots))))$$

$${}_l[\mathbf{F} \mathbf{G} p]_k^0 = (L_p(z_0) \wedge L_p(z_1) \wedge \dots \wedge L_p(z_k)) \vee \\ (L_p(z_1) \wedge \dots \wedge L_p(z_k)) \vee \dots \vee \\ (L_p(z_l) \wedge \dots \wedge L_p(z_k))$$

Rozmiar formuły boolowskiej $l[\phi]_k^0$ jest $\mathcal{O}(k \cdot |\phi|)$

dzięki „dzieleniu podformuł” $l[\phi]_k^i$

$$[\phi]_k^i \quad 0 \leq i \leq k$$

$$s_0 \longrightarrow s_1 \longrightarrow \dots \longrightarrow s_k$$

$$[p]_k^i \quad [\neg p]_k^i \quad [\phi \wedge \psi]_k^i \quad [\phi \vee \psi]_k^i \quad := \quad \text{j. w.}$$

$$[\mathbf{X} \phi]_k^i \quad := \quad [\phi]_k^{i+1}$$

$$[\phi]_k^{k+1} \quad := \quad \text{false}$$

$$\mathbf{F} \phi \quad \equiv \quad \phi \vee \mathbf{X} \mathbf{F} \phi$$

$$\phi \mathbf{U} \psi \quad \equiv \quad \psi \vee (\phi \wedge \mathbf{X} \phi \mathbf{U} \psi)$$

$$\mathbf{G} \phi \quad \equiv \quad \phi \wedge \mathbf{X} \mathbf{G} \phi$$

$$\phi \mathbf{R} \psi \quad \equiv \quad \psi \wedge (\phi \vee \mathbf{X} \phi \mathbf{R} \psi)$$

Przykład:

$$s_0 \longrightarrow s_1 \longrightarrow \dots \longrightarrow s_k$$

$$[\mathbf{F} p]_k^0 = L_p(z_0) \vee \dots \vee L_p(z_k) \vee \mathbf{false}$$

$$[\mathbf{G} p]_k^0 = L_p(z_0) \wedge \dots \wedge L_p(z_k) \wedge \mathbf{false} \equiv \mathbf{false}$$

$$[p \mathbf{U} q]_k^0 = L_q(z_0) \vee (L_p(z_0) \wedge (L_q(z_1) \vee (\dots \\ L_q(z_{k-1}) \vee (L_p(z_{k-1}) \wedge (L_q(z_k) \vee (L_p(z_k) \wedge \mathbf{false})))) \dots)))$$

$$[\mathbf{F} \mathbf{G} p]_k^0 \equiv \mathbf{false}$$

$$[M, \phi]_k := [M]_k \wedge \left((\neg L_k \wedge [\phi]_k^0) \vee \bigvee_{l=0}^k ({}_l L_k \wedge {}_l [\phi]_k^0) \right)$$

Tw.: $M \models_k \mathbf{E} \phi \iff [M, \phi]_k$ jest spełnialna.

Powtarzaj:

(1) $k := k_0$

(2) jeśli $[M, \phi]_k$ spełnialna to stop

$M \models \mathbf{E} \phi$

(3) zwiększ k i kontunuuj

Inne kodowania boolowskie

- weryfikacja oprogramowania (np. CBMC)
- weryfikacji sprzętu:
 - zachowana jest informacja o strukturze układu
 - desygnowane SAT-rozwiązywacze
- LTL z przeszłością
- programy współbieżne
- ...

II. Pełność ?

Jak uzyskać pełność weryfikacji ograniczonej?

- ograniczenie dla k (bezpieczeństwo)
- równolegle sprawdzamy ϕ i $\neg\phi$ (żywołność)
- indukcja (bezpieczeństwo)

Głębokość – ograniczenie dla k

najmniejsze i t. że

$$\forall z_0, \dots, z_n. \exists z'_0, \dots, z'_t, t \leq i. S_0(z_0) \wedge \bigwedge_{j=0}^{n-1} R(z_j, z_{j+1}) \implies$$

$$S_0(z'_0) \wedge \left(\bigwedge_{j=0}^{t-1} R(z'_j, z'_{j+1}) \right) \wedge z'_t = z_n$$

Głębokość – ograniczenie dla k

najmniejsze i t. że

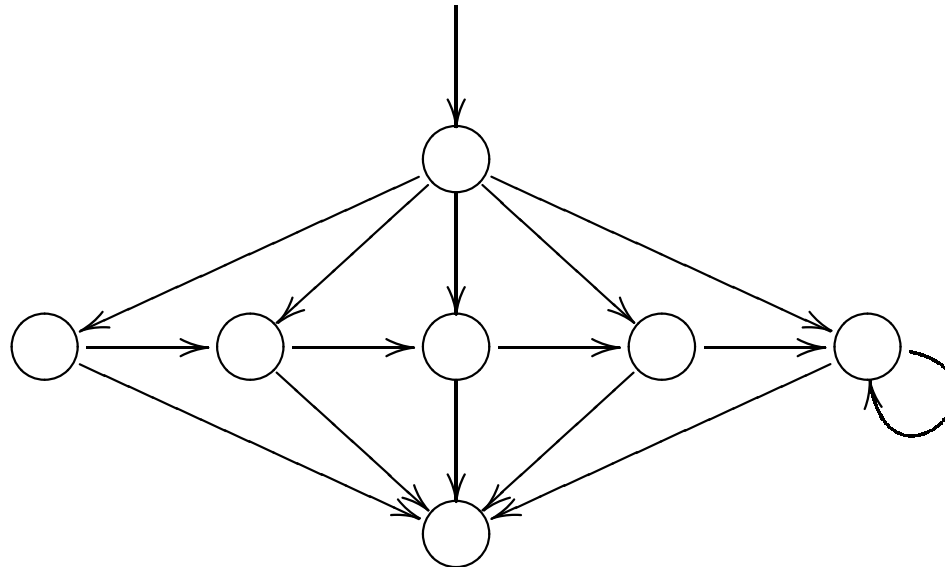
$$\forall z_0, \dots, z_{i+1}. \exists z'_0, \dots, z'_i. S_0(z_0) \wedge \bigwedge_{j=0}^i R(z_j, z_{j+1}) \implies$$

$$S_0(z'_0) \wedge \left(\bigwedge_{j=0}^{i-1} R(z'_j, z'_{j+1}) \right) \wedge \bigvee_{j=0}^i z'_j = z_{i+1}$$

najdłuższa ścieżka bez pętli – ograniczenie dla k

największe i t. że

$$\exists z'_0, \dots, z_i. S_0(z_0) \wedge \left(\bigwedge_{j=0}^{i-1} R(z_j, z_{j+1}) \right) \wedge \bigwedge_{j=0}^{i-1} \bigwedge_{l=j+1}^i z_j \neq z_l$$



$M \models \mathbf{EG} \neg p \iff \exists k$ t. że następująca formuła jest spełnialna:

$$S_0(z_0) \wedge \bigwedge_{j=0}^{k-1} R(z_j, z_{j+1}) \wedge \bigvee_{l=0}^k ({}_l L_k \wedge \bigwedge_{j=0}^k \neg L_p(z_j))$$

$M \models \mathbf{AF} p \iff \exists k$ t. że następująca formuła jest tautologią:

$$S_0(z_0) \wedge \bigwedge_{j=0}^{k-1} R(z_j, z_{j+1}) \implies \bigvee_{j=0}^k L_p(z_j)$$

$M \models \mathbf{EG} \neg p \iff \exists k$ t. że następująca formuła jest spełnialna:

$$S_0(z_0) \wedge \bigwedge_{j=0}^{k-1} R(z_j, z_{j+1}) \wedge \left(\bigvee_{l=0}^k lL_k \right) \wedge \bigwedge_{j=0}^k \neg L_p(z_j)$$

$M \models \mathbf{AF} p \iff \exists k$ t. że następująca formuła jest niespełnialna:

$$S_0(z_0) \wedge \bigwedge_{j=0}^{k-1} R(z_j, z_{j+1}) \wedge \bigwedge_{j=0}^k \neg L_p(z_j)$$

- warunki początkowe

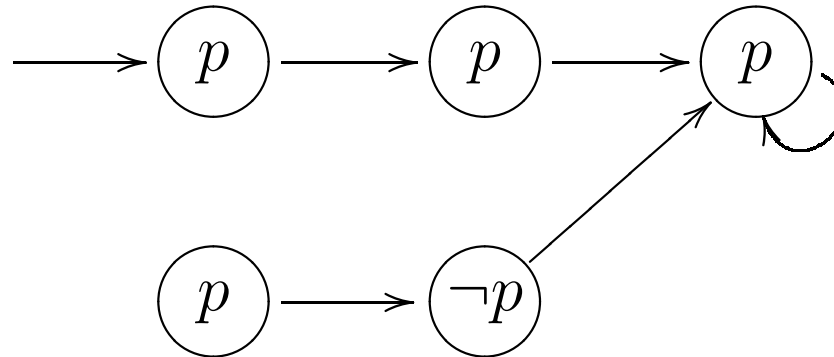
sprawdzamy, że poniższa formuła jest niespełnialna:

$$\text{Base}_k(z_0, \dots, z_k) \equiv S_0(z_0) \wedge \bigwedge_{j=0}^{k-1} R(z_j, z_{j+1}) \wedge \bigvee_{j=0}^k \neg L_p(z_j)$$

- krok indukcyjny

sprawdzamy, że poniższa formuła jest niespełnialna:

$$\text{Step}_k(z_0, \dots, z_{k+1}) \equiv \bigwedge_{j=0}^k (L_p(z_j) \wedge R(z_j, z_{j+1})) \wedge \neg L_p(z_{k+1})$$



Powtarzaj:

(1) $k := 0$

(2) jeśli Base_k spełnialna to stop

$$M \models \text{EF } \neg \phi$$

(3) jeśli Step_k niespełnialna to stop

$$M \models \text{G } p$$

(4) zwiększ k i kontunuuj

III. BDD czy SAT ?

bit	k	SMV ₂	MB	PROVER	MB
0	1	25	79	< 1	1
1	2	25	79	< 1	1
2	3	26	80	< 1	1
3	4	27	82	1	2
4	5	33	92	1	2
5	6	67	102	1	2
6	7	258	172	2	2
7	8	1741	492	7	3
8	9		> 1GB	29	3
9	10			58	3
10	11			91	3
11	12			125	3
12	13			156	4
13	14			186	4
14	15			226	4
15	16			183	5

[Biere, Cimatti, Clarke, Strichman, Zhu 2003]

- metody są komplementarne
- SAT bardziej przewidywalny
- nieograniczona weryfikacja modelowa UMC:
CNF zamiast BDD w SMC
- BDD + SAT

IV. SAT-rozwiązywacze

- CNF
- algorytm DPLL
 - przeszukiwanie drzewa wartościowań częściowych
 - BCP (ang. *Boolean Constraint Propagation*)
 - **konflikty** – „obcinanie” drzewa
- heurystyki

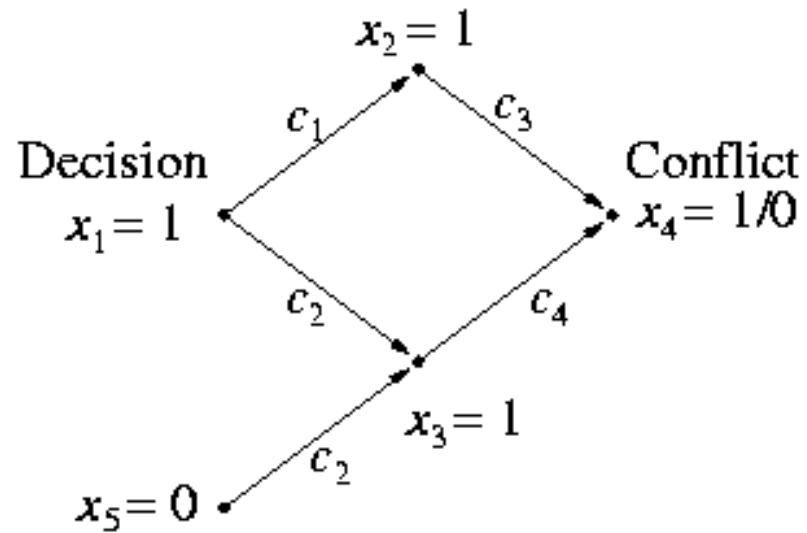
Graf implikacije i BCP

$$c_1 = (\neg x_1 \vee x_2)$$

$$c_2 = (\neg x_1 \vee x_3 \vee x_5)$$

$$c_3 = (\neg x_2 \vee x_4)$$

$$c_4 = (\neg x_3 \vee \neg x_4)$$



[Biere, Cimatti, Clarke, Strichman, Zhu 2003]

```

// Input arg: Current decision level  $d$ 
// Return value:
//   SAT():      {SAT, UNSAT}
//   Decide():   {DECISION, ALL-DECIDED}
//   Deduce():   {OK, CONFLICT}
//   Diagnose(): {SWAP, BACK-TRACK} also calculates  $\beta$ 

```

```

SAT (d)
{
  l1:   if (Decide (d) == ALL-DECIDED) return SAT;
  l2:   while (TRUE) {
  l3:     if (Deduce(d) != CONFLICT) {
  l4:       if (SAT (d+1) == SAT) return SAT;
  l5:       else if ( $\beta < d$  ||  $d == 0$ )
  l6:         { Erase (d); return UNSAT; }
        }
  l7:   if (Diagnose (d) == BACK-TRACK) return UNSAT;
}
}

```

[Biere, Cimatti, Clarke, Strichman, Zhu 2003]

Dlaczego SAT-rozwiązywacze są tak szybkie?

- uczenie konfliktów – dodajemy tzw. klauzule konfliktowe
- niechronologiczne powroty
- heurystyki dla:
 - decyzji
 - ...
- szybkie struktury danych
- inkrementalna spełnialność