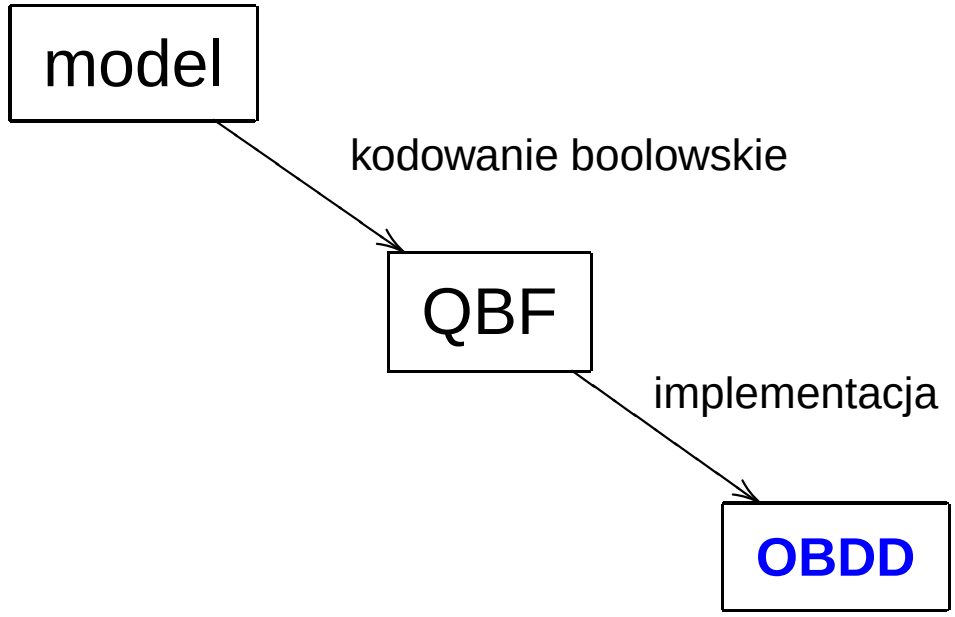
Praktyczne metody weryfikacji

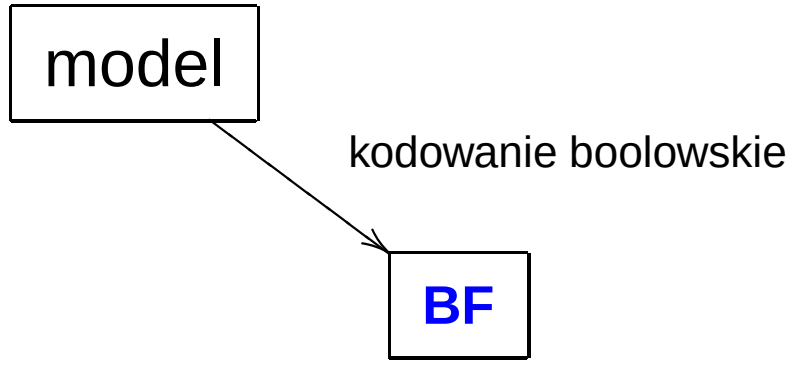
Wykład 9: Weryfikacja ograniczona

Symboliczna weryfikacja modelowa (SMC)



weryfikacja modelowa = operacje na OBDDs

Ograniczona weryfikacja modelowa (BMC)



weryfikacja modelowa = spełnialność formuły boolowskiej

Przykł ad: EF $\neg p$ (bezpieczeństwo)

pre-SMC:

$$\ldots \subseteq \neg p \cup \mathsf{EX}(\neg p \cup \mathsf{EX}\neg p) \supseteq \neg p \cup \mathsf{EX}\neg p \supseteq \neg p \supseteq \texttt{false}$$

$$\ldots \subseteq \neg p \cup \mathrm{pre}(\neg p \cup \mathrm{pre}(\neg p)) \supseteq \neg p \cup \mathrm{pre}(\neg p) \supseteq \neg p \supseteq \texttt{false}$$

post-SMC:

$$S_0 \subseteq S_0 \cup \mathrm{post}(S_0) \subseteq S_0 \cup \mathrm{post}(S_0 \cup \mathrm{post}(S_0)) \subseteq \ldots$$

Ograniczona weryfikacja modelowa

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

I. Weryfikacja ograniczona

$$M \models \mathsf{E}\,\phi \qquad (\phi \in \mathsf{LTL}^{+})$$

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

M opisany przez formuł y boolowskie:

- $\widehat{R}(x_1, \ldots, x_m, x'_1, \ldots, x'_m)$
- $\widehat{L_p}(x_1, \ldots, x_m)$, $\widehat{S}_0(x_1, \ldots, x_m)$

Pomysł : horyzont $k \geq 0$ k kroków systemu

$\Pi \models_k \phi$ k kroków wystarcza by stwierdzić, że $\Pi \models \phi$

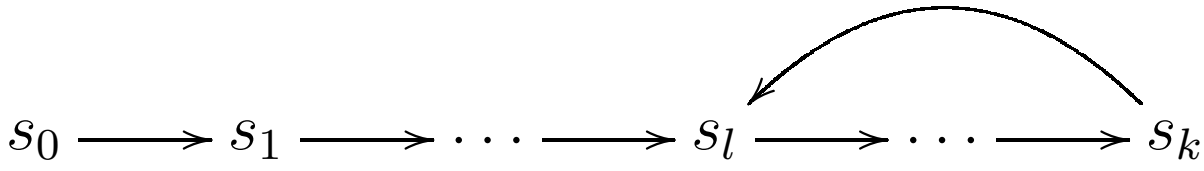
Lem.: $\Pi \models_k \phi \implies \Pi \models \phi$

Lem.: $M \models \mathsf{E}\,\phi \implies \exists k \geq 0.\ M \models_k \mathsf{E}\,\phi$

Tw.: $M \models \mathsf{E}\,\phi \iff \exists k \geq 0.\ M \models_k \mathsf{E}\,\phi$

$$\Pi \quad = \quad s_0 \to s_1 \to s_2 \to \dots$$

(k, l) - pętla:



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

(k - pętla)

brak pętli:

Semantyka ograniczona dla k-pętli

$$\Pi \models_k \phi \iff \Pi \models \phi \qquad \Pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow \ldots$$

Semantyka ograniczona dla nie-pętli

$$\Pi \models_k \phi \iff \Pi \models_k^0 \phi \qquad \Pi = s_0 \to s_1 \to s_2 \to \ldots$$

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

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

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

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

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

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

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

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

$$\phi \,\mathsf{R}\, \psi \;\equiv\; \neg(\neg\phi U \neg\psi)$$

$$\Pi \vDash \phi \,\mathsf{R}\, \psi \;\text{ wtw gdy }\; \forall i < |\Pi|.\; (\forall j < i.\; \Pi^j \vDash \neg\phi) \implies \Pi^i \vDash \psi$$

$$\phi \,\mathsf{R}\, \psi \;\equiv\; \psi \,\mathsf{U}\, (\psi \wedge \phi) \;\vee\; \mathsf{G}\, \psi$$

$$\Pi \;\vDash^i_k \phi \,\mathsf{U}\, \psi \iff \exists j,\; i \leq j \leq k.\; \Pi \;\vDash^j_k \psi \;\wedge\; \forall l,\; i \leq l < j.\; \Pi \;\vDash^l_k \phi$$

$$\Pi \;\vDash^i_k \phi \,\mathsf{R}\, \psi \iff \exists j,\; i \leq j \leq k.\; \Pi \;\vDash^j_k \phi \;\wedge\; \forall l,\; i \leq l \leq j.\; \Pi \;\vDash^l_k \psi$$

Lem.: $\Pi \models_k \phi \implies \Pi \models \phi$

Lem.: $M \models \mathsf{E}\,\phi \implies \exists k \geq 0.\ M \models_k \mathsf{E}\,\phi$

Tw.: $M \models \mathsf{E}\,\phi \iff \exists k \geq 0.\ M \models_k \mathsf{E}\,\phi$

- **jakie k jest wystarczająco duże? średnica grafu?**
- **brak pełności !**

$$M \models_k \mathsf{E}\,\phi \qquad (\phi \in \mathsf{LTL}^+)$$

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

$M \models_k \phi \iff [M, \phi]_k$ spełnialna

M opisany przez formuły boolowskie:

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

Ścieżka symboliczna

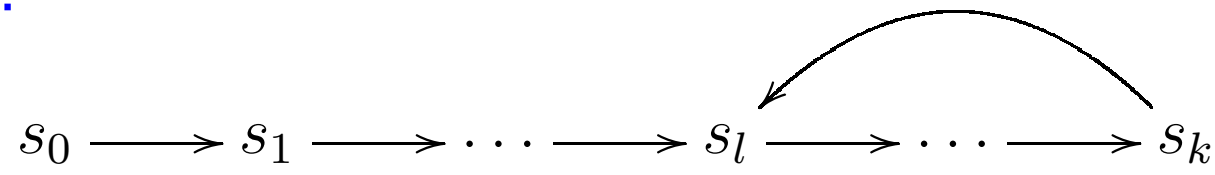
„$z_0 \longrightarrow z_1 \longrightarrow \ldots \longrightarrow z_k$”

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

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

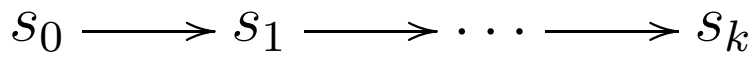
rozmiar formuły boolowskiej $[M]_k$ jest $\mathcal{O}(k \cdot |M|)$

(k, l) - pętla:



$$ {}_lL_k \equiv R(z_k, z_l) \qquad\qquad L_k \equiv \bigvee_{l=0}^{k} {}_lL_k $$

brak pętli:



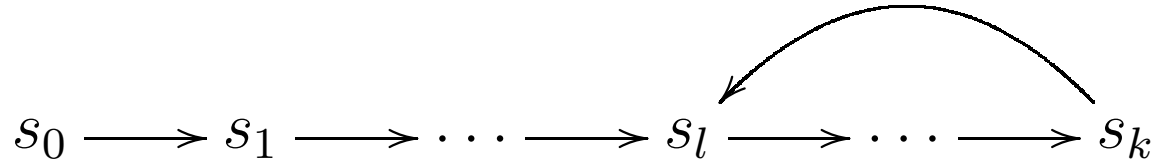
$$[M,\phi]_k \;:=\; [M]_k \;\wedge\; \Big((\neg L_k \wedge [\phi]^0_k) \vee \bigvee_{l=0}^{k} ({}_lL_k \wedge {}_l[\phi]^0_k)\Big)$$

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

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

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

$_l[\phi]^i_k \qquad 0 \leq l, i \leq k$



$$_l[p]^i_k := L_p(z_i)$$

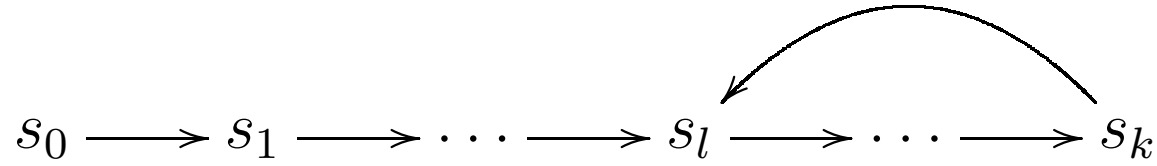
$$_l[\neg p]^i_k \qquad _l[\phi \wedge \psi]^i_k \qquad _l[\phi \vee \psi]^i_k := \ldots$$

$$_l[\mathsf{X}\,\phi]^i_k := {}_l[\phi]^{\mathrm{succ}(i)}_k \qquad \mathrm{succ}(i) = \begin{cases} i+1 & i < k \\ l & i = k \end{cases}$$

$$\mathsf{F}\,\phi \equiv \phi \vee \mathsf{X}\,\mathsf{F}\,\phi \qquad \phi\,\mathsf{U}\,\psi \equiv \psi \vee (\phi \wedge \mathsf{X}\,\phi\,\mathsf{U}\,\psi)$$

$$\mathsf{G}\,\phi \equiv \phi \wedge \mathsf{X}\,\mathsf{G}\,\phi \qquad \phi\,\mathsf{R}\,\psi \equiv \psi \wedge (\phi \vee \mathsf{X}\,\phi\,\mathsf{R}\,\psi)$$

Przykł ad:



$$_l[\,\mathsf{F}\,p]_k^0 \;=\; L_p(z_0) \vee \ldots \vee L_p(z_k)$$

$$_l[\,\mathsf{G}\,p]_k^0 \;=\; L_p(z_0) \wedge \ldots \wedge L_p(z_k)$$

$$_l[p\,\mathsf{U}\,q]_k^0 \;=\; L_q(z_0) \vee (L_p(z_0) \wedge (L_q(z_1) \vee (\ldots$$

$$L_q(z_{k-1}) \vee (L_p(z_{k-1}) \wedge L_q(z_k))\ldots)))$$

$$_l[\,\mathsf{F}\;\mathsf{G}\,p]_k^0 \;=\; (L_p(z_0) \wedge L_p(z_1) \wedge \ldots \wedge L_p(z_k)) \vee$$

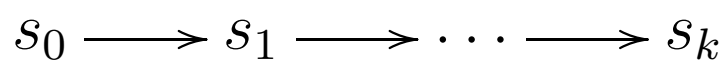
$$(L_p(z_1) \wedge \ldots \wedge L_p(z_k)) \vee$$

$$L_p(z_k)$$

Rozmiar formuł y boolowskiej ${}_l[\phi]_k^i$ jest $\mathcal{O}(k \cdot |\phi|)$

dzięki „dzieleniu podformuł ”

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



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

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

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

$$\mathsf{F}\,\phi \equiv \phi \vee \mathsf{X}\,\mathsf{F}\,\phi \qquad \phi\,\mathsf{U}\,\psi \equiv \psi \vee (\phi \wedge \mathsf{X}\,\phi\,\mathsf{U}\,\psi)$$

$$\mathsf{G}\,\phi \equiv \phi \wedge \mathsf{X}\,\mathsf{G}\,\phi \qquad \phi\,\mathsf{R}\,\psi \equiv \psi \wedge (\phi \vee \mathsf{X}\,\phi\,\mathsf{R}\,\psi)$$

Przykł ad:

$$s_0 \longrightarrow s_1 \longrightarrow \cdots \longrightarrow s_k$$

$$[\,\mathsf{F}\,p]_k^0 \;=\; L_p(z_0) \vee \ldots \vee L_p(z_k) \vee \texttt{false}$$

$$[\,\mathsf{G}\,p]_k^0 \;=\; L_p(z_0) \wedge \ldots \wedge L_p(z_k) \wedge \texttt{false} \;\equiv\; \texttt{false}$$

$$[p\,\mathsf{U}\,q]_k^0 \;=\; L_q(z_0) \vee (L_p(z_0) \wedge (L_q(z_1) \vee (\ldots$$

$$L_q(z_{k-1}) \vee (L_p(z_{k-1}) \wedge (L_q(z_k) \vee (L_p(z_k) \wedge \texttt{false}))) \ldots))))$$

$$[\,\mathsf{F}\;\mathsf{G}\,p]_k^0 \;\equiv\; \texttt{false}$$

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

Tw.: $M \models_k \mathsf{E}\,\phi \iff [M,\phi]_k$ jest speł nialna.

II. Pełność ?

Jak uzyskać pełność weryfikacji ograniczonej?

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

Średnica – ograniczenie dla k

najmniejsze i t. że

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

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

Średnica – ograniczenie dla k

najmniejsze i t. że

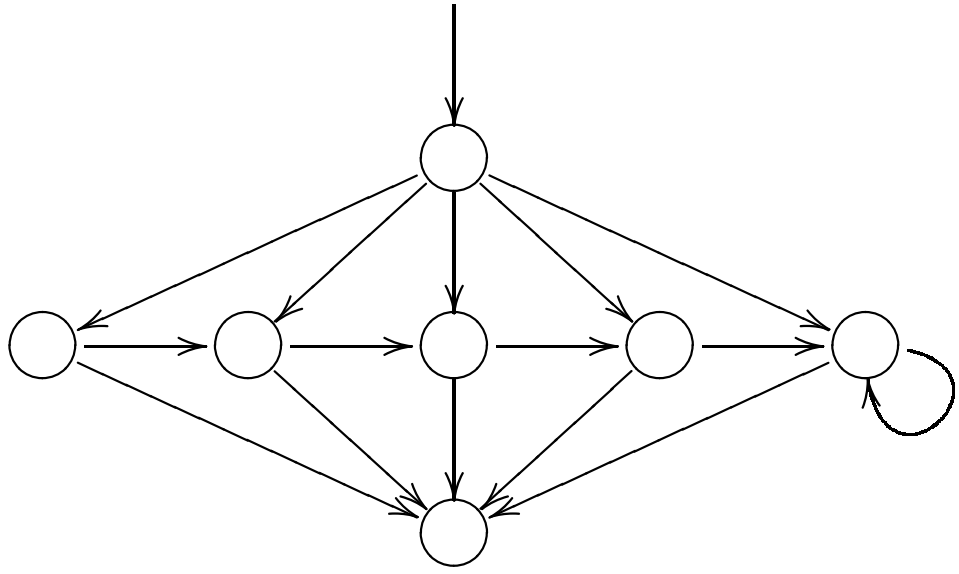
$$\forall z_0, \ldots, z_{i+1}.\ \exists z'_0, \ldots, z'_i.\ S_0(z_0)\ \wedge\ \bigwedge_{j=0}^{i} R(z_j, z_{j+1})\ \Longrightarrow$$

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

najdłuższa ścieżka bez pętli – ograniczenie dla k

największe i t. że

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



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

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

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

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

Pełność dla $\mathsf{F}\,p$

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

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

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

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

Pełność dla $\mathbf{G}\,p$ – indukcja

1) wybieramy niezmiennik $\phi(z)$

2) sprawdzamy, że poniższa formuła jest niespełnialna:

$$\exists z_0, \ldots, z_k.\ S_0(z_0) \ \wedge \ \bigwedge_{j=0}^{k-1} R(z_j, z_{j+1}) \ \wedge \ \bigvee_{j=0}^{k} \neg\phi(z_j)$$

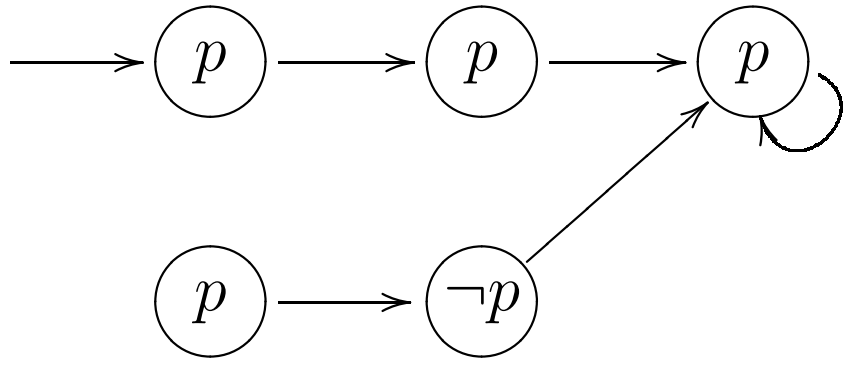
3) sprawdzamy, że poniższa formuła jest niespełnialna:

$$\exists z_0, \ldots, z_{k+1}.\ \bigwedge_{j=0}^{k} (\phi(z_j) \wedge R(z_j, z_{j+1})) \ \wedge \ \neg\phi(z_{k+1})$$

4) sprawdzamy, że poniższa formuła jest tautologią:

$$\forall z.\ \phi(z) \implies L_p(z)$$

Po co $k > 1$?



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

– BDD + SAT

– nieograniczona weryfi kacja modelowa

IV. SAT-solvers

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

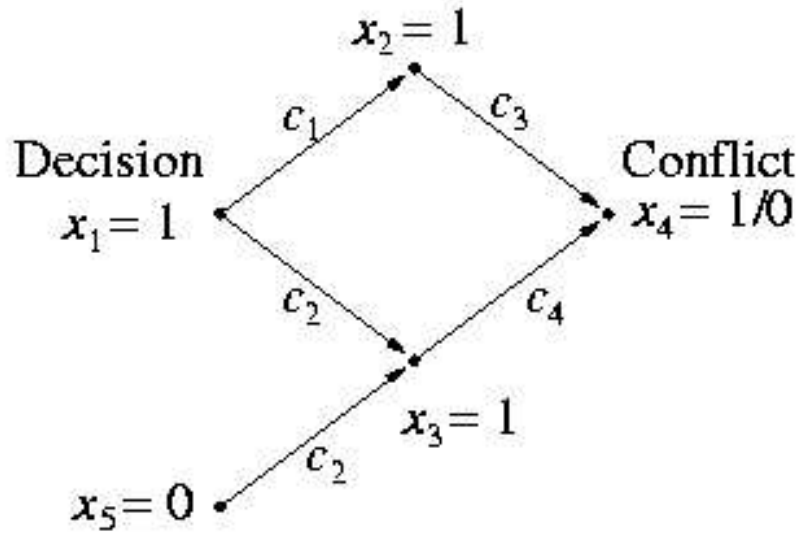
Graf implikacji

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

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

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

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



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

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 (β < d || d == 0)
l6:             { Erase (d); return UNSAT; }
          }
l7:       if (Diagnose (d) == BACK-TRACK) return UNSAT;
       }
   }
```

Dlaczego SAT-rozwiązywacze są tak szybkie?

- uczenie konfliktów – dodajemy tzw. klauzule konfliktowe
- niechronologiczne powroty
- heurystyki dla:
  - decyzji
  - …