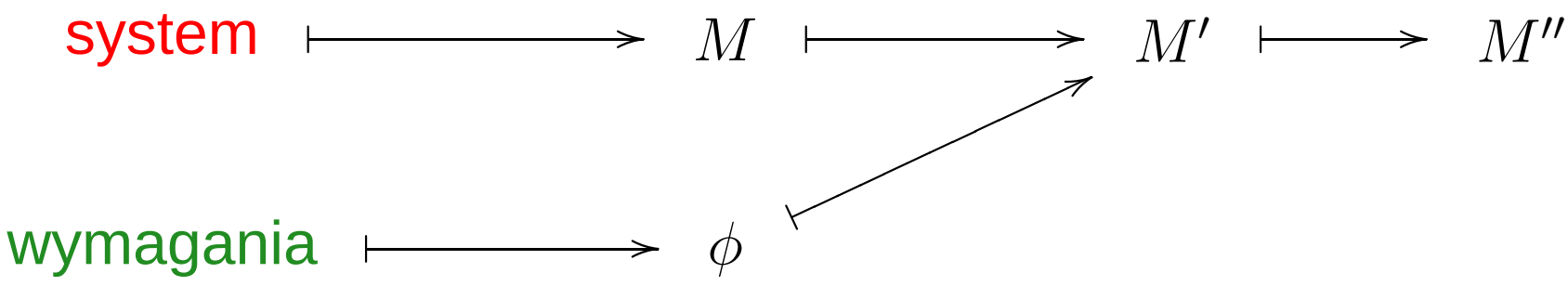
Praktyczne metody weryfikacji

Wykład 10: Abstrakcja

Od rzeczywistości do modelu



$M'' \models \phi$?

Abstrakcja = usuwanie z modelu

zbędnej informacji

Metody abstrakcji:

- eliminacja zmiennych
- abstrakcja danych
- abstrakcja oparta na symetrii
- ...

$$M \longmapsto M'$$

Model: $M = (S, S_0, L, R)$

$$V = \{x_1, \ldots, x_n\}, \qquad S = D^n$$

Uwaga: abstrakcja dotyczy opisu modelu

R, S_0, L reprezentowane przez formuły

I. Eliminacja zmiennych

Bisymulacja

$B \subseteq S \times S'$

$(s, s') \in B \implies$

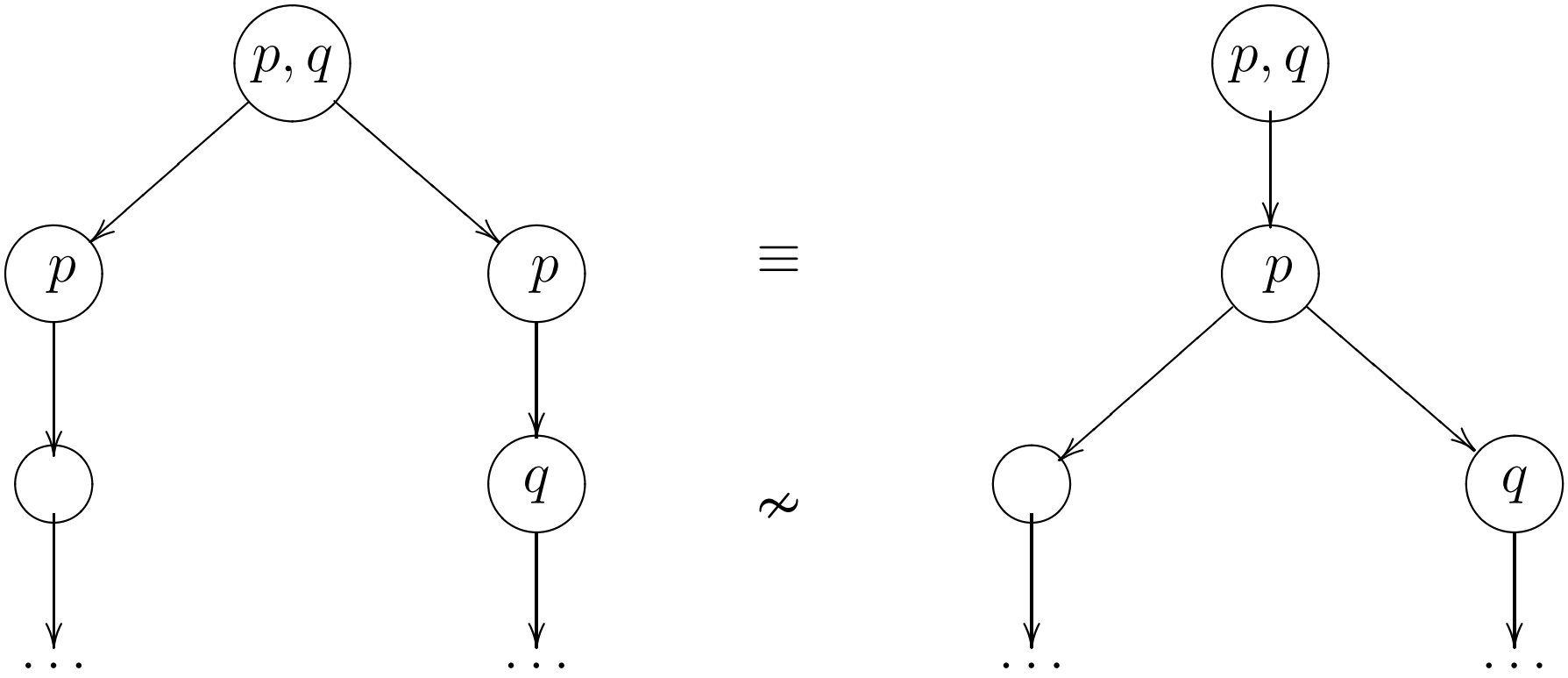
- $L(s) = L(s')$
- $s \to r \implies \exists r'.\ s' \to r',\ (r, r') \in B$
- $s' \to r' \implies \exists r.\ s \to r,\ (r, r') \in B$

$s \sim s' \iff (s, s') \in B$ dla pewnej bisymulacji B

$M \sim M' \iff s_0 \sim s'_0$

gra $G_{M\ M'}$

Przykład



Bisymulacja: gra $G_{M\ M'}$

– gracze: Spoiler, Duplikator

– konfiguracje: $\langle s, s' \rangle$ – konfiguracja pocz.: $\langle s_0, s'_0 \rangle$

– jeśli $L(s) \neq L(s')$, to wygrywa Spoiler

– gracz I wybiera stronę i tranzycję $s \longrightarrow r$ (lub $s' \longrightarrow r'$)

– gracz II odpowiada po drugiej stronie

– itd.

– kto wygrywa?

Bisymulacja

Tw.: $M \sim M' \implies (\forall \phi \in \mathrm{CTL}^*.\ M \models \phi \iff M' \models \phi)$

Tw.: $M \sim M' \impliedby (\forall \phi \in \mathrm{CTL}.\ M \models \phi \iff M' \models \phi)$

Wn.: $\equiv_{\mathrm{CTL}^*} = \equiv_{\mathrm{CTL}} = \ldots = \equiv_{\mathsf{EX}}$

Algorytm symboliczny:

$$B_0 := \{\langle s, s' \rangle \mid L(s) = L'(s')\}$$

powtarzaj

$$B_{i+1} := \{\langle s, s' \rangle \mid s \to r \implies \exists r'.\, s' \to r',\ (r, r') \in B_i \wedge$$

$$s' \to r' \implies \exists r.\, s \to r,\ (r, r') \in B_i\}$$

aż $B_{i+1} = B_i$

Sprawiedliwa bisymulacja

$(s, s') \in B \implies$

- $L(s) = L(s')$
- $\forall$ sprawiedl. Π z s $\exists$ sprawiedl. Π' z s'. $(\Pi, \Pi') \in B$
- $\forall$ sprawiedl. Π' z s' $\exists$ sprawiedl. Π z s. $(\Pi, \Pi') \in B$

$s \sim_F s' \iff (s, s') \in B$ dla pewnej sprawiedl. bisymulacji B

$M \sim_F M' \iff s_0 \sim_F s'_0$

Eliminacja zmiennych

Def: $V' \subseteq V$; $C(V') \subseteq V$ zmienne od których zależy V'

Przykład: licznik 3-bitowy

$$\begin{aligned} R_0(\vec{x}, x'_0) &= (x'_0 = \neg x_0) \\ R_1(\vec{x}, x'_1) &= (x'_1 = x_0 \text{ xor } x_1) \\ R_2(\vec{x}, x'_2) &= (x'_2 = (x_0 \wedge x_1) \text{ xor } x_2) \end{aligned}$$

$C(\{x_i\}) = \{x_0, \ldots, x_i\}$

Eliminacja zmiennych

- $\{x_1, \ldots, x_k\} := C(V')$, V' - zmienne w formule
- $S' = D^k$
- $R'(x_1, \ldots, x_k, x'_1, \ldots, x'_k)$
- $S'_0(x_1, \ldots, x_k) \equiv \exists x_{k+1}, \ldots, x_n.\ S_0(x_1, \ldots, x_n)$

$$(d_1, \ldots, d_n)B(d'_1, \ldots, d'_k) \iff d_1 = d'_1 \wedge \ldots \wedge d_k = d'_k$$

II. Abstrakcja danych

Symulacja

$P \supseteq P'$

$(s, s') \in S \implies$

- $L(s) \cap P' = L(s')$
- $s \to r \implies \exists r'.\ s' \to r',\ (r, r') \in S$

$s \preceq s' \iff (s, s') \in S$ dla pewnej symulacji S

$M \preceq M' \iff s_0 \preceq s'_0$

Symulacja

Tw.: $M \preceq M' \implies (\forall \phi \in \mathrm{ACTL}^*.\ M' \models \phi \implies M \models \phi)$

Tw.: $M \preceq M' \implies (\forall \phi \in \mathrm{ECTL}^*.\ M \models \phi \implies M' \models \phi)$

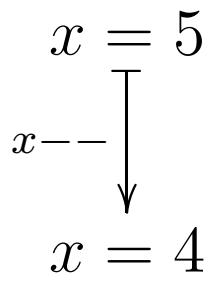
Def.: $\simeq\ =\ \preceq \cap \succeq$

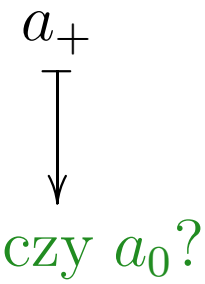
Pytanie: Czy $\sim\ =\ \simeq$?

$\preceq_F$, $\simeq_F$ analogicznie

Przykład:

$$h_x(d) = \begin{cases} a_+ & x > 0 \\ a_0 & x = 0 \\ a_- & x < 0 \end{cases} \qquad h_x : D \to A$$





Przykład:

$$h_x(d) = \begin{cases} a_+ & x > 0 \\ a_0 & x = 0 \\ a_- & x < 0 \end{cases} \qquad h_x : D \to A$$

– model abstrakcyjny ma więcej zachowań – symulacja

– niedeterminizm

Model ilorazowy

$$M \mapsto \widehat{M} = M/\approx_h$$

$$\approx_h = \ker(h)$$

$$R(\vec{d}_1, \vec{d}_2) \Longrightarrow \widehat{R}([\vec{d}_1], [\vec{d}_2])$$

$$S_0(\vec{d}) \Longrightarrow \widehat{S}_0([\vec{d}])$$

Tw.: $M \preceq \widehat{M}$

Tw.: $M \sim \widehat{M}$ gdy $\approx_h$ jest bisymulacją

Przykładowe funkcje abstrakcji:

- znak
- logarytm
- wybrany bit
- kongruencja $\mod m$
- $h = \langle h_1, h_2 \rangle$
- predykaty ($\rightsquigarrow$ programy „boolowskie")
- ...

Abstrakcja egzystencjalna

$$h_x : D \to A$$

- $\widehat{S} = A^n$

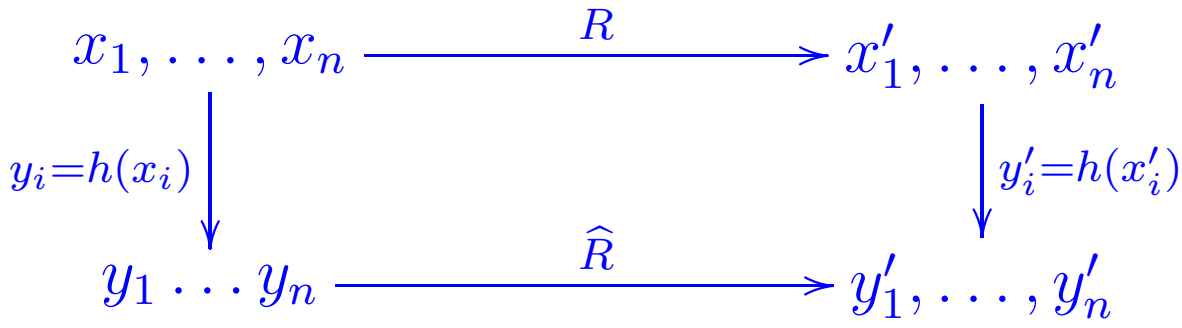
$$R(\vec{d_1}, \vec{d_2}) \implies \widehat{R}([\vec{d_1}], [\vec{d_2}])$$

$$S_0(\vec{d}) \implies \widehat{S_0}([\vec{d}])$$

Abstrakcja egzystencjalna

$$h_x : D \to A$$

- $\widehat{S} = A^n$
- $\widehat{R}(y_1, \ldots, y_n, y'_1, \ldots, y'_n) \equiv \exists x_1 \ldots x_n, x'_1, \ldots, x'_n.$
  $R(x_1, \ldots, x_n, x'_1, \ldots, x'_n) \wedge \bigwedge_i y_i{=}h(x_i) \wedge y'_i{=}h(x'_i)$

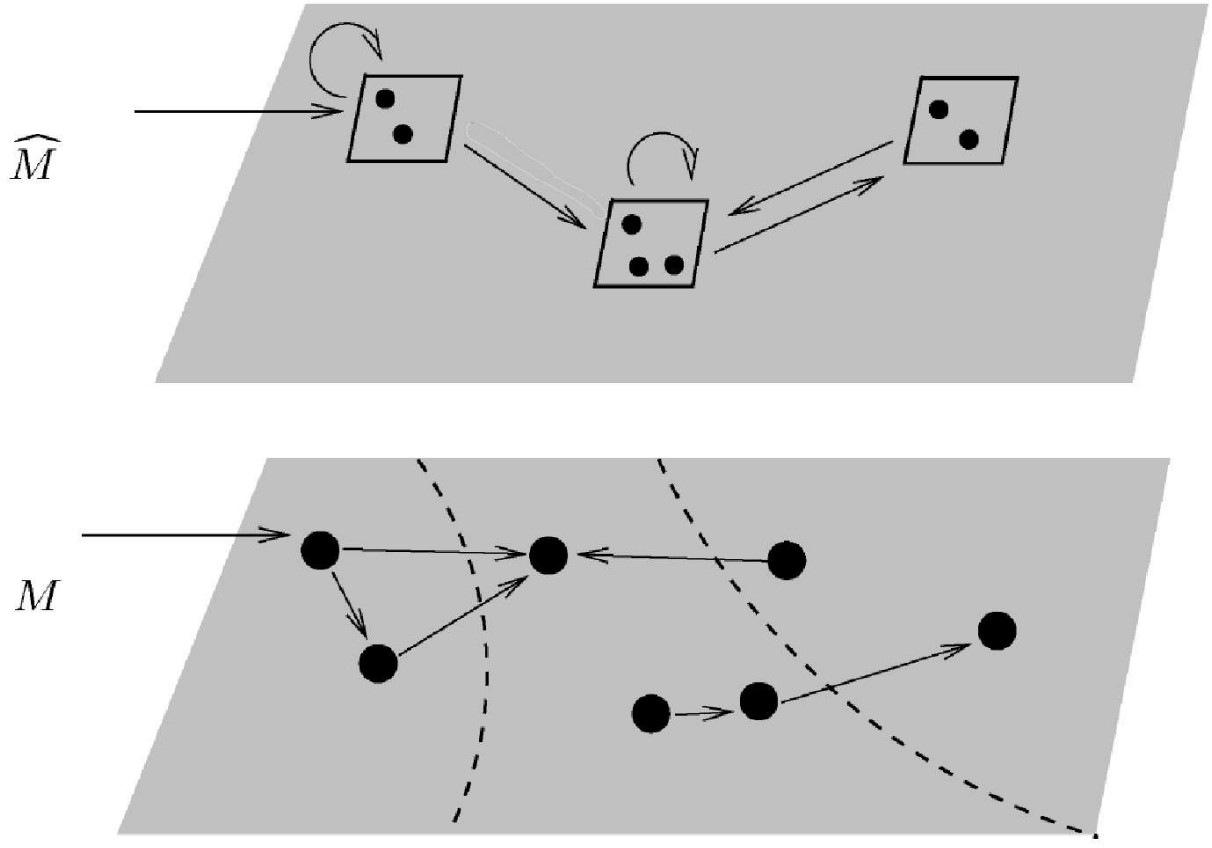


- $\widehat{S}_0(y_1, \ldots, y_n) \equiv \exists x_1 \ldots x_n.\ S_0(x_1, \ldots, x_n) \wedge \bigwedge_i y_i{=}h(x_i)$

$[\phi](\vec{y}) \equiv \exists \vec{x}.\ \phi(\vec{x}) \wedge \bigwedge_i y_i{=}h(x_i)$

Abstrakcja egzystencjalna

$$\widehat{R}(y_1,\ldots,y_n,y'_1,\ldots,y'_n) \equiv \exists x_1\ldots x_n,x'_1,\ldots,x'_n.$$
$$R(x_1,\ldots,x_n,x'_1,\ldots,x'_n) \wedge \bigwedge_i y_i{=}h(x_i) \wedge y'_i{=}h(x'_i)$$



[Clarke, Grumberg, Jha, Lu, Veith 2003]

Abstrakcja egzystencjalna – przybliżenie

Obliczamy przybliżenie $[\phi]$

- $\mathcal{A}(\phi_1 \wedge \phi_2) \;=\; \mathcal{A}(\phi_1) \wedge \mathcal{A}(\phi_2)$
- $\ldots$
- $\mathcal{A}(\exists x.\, \phi) \;=\; \exists x.\, \mathcal{A}(\phi)$
- $\mathcal{A}(x' = x{-}1) \;=\; (y = a_+ \wedge (y' = a_+ \vee y' = a_0)) \vee \ldots$

Fakt: $[\phi] \implies \mathcal{A}(\phi)$

Tw.: $M \preceq \widehat{M} \preceq M^{\mathcal{A}}$

III. CEGAR

Uszczegółowienie abstrakcji – CEGAR

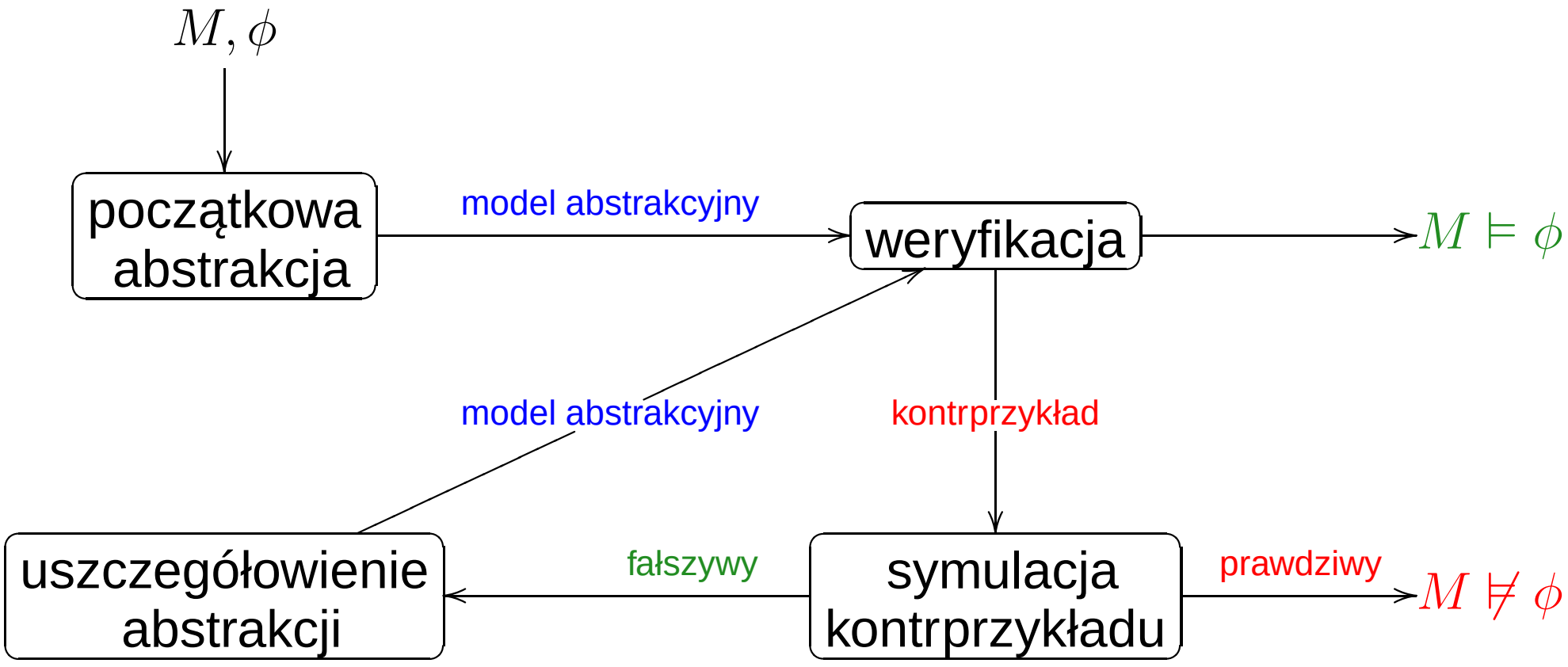
(ang. Counter-example guided abstraction refinement)

(1) Znajdź abstrakcję początkową

(2) Powtarzaj:

- weryfikuj model abstrakcyjny
  - jeśli wynik pozytywny – zakończ
  - jeśli wynik negatywny – kontrprzykład abstrakcyjny
- jesli kontrprzykład jest rzeczywisty – zakończ
- uszczegółów abstrakcję na podstawie kontrprzykładu

CEGAR


M, ϕ
początkowa abstrakcja
model abstrakcyjny
weryfikacja
$M \models \phi$
model abstrakcyjny
kontrprzykład
uszczegółowienie abstrakcji
fałszywy
symulacja kontrprzykładu
prawdziwy
$M \not\models \phi$

CEGAR: abstrakcja początkowa

init(y) := 1;
next(y) := **case**
$(r = 1) : 0;$
$(x < y) \wedge \neg(y = 2) : y + 1;$
$(x = y) : 0;$
else : $y;$
esac

predykaty:
$(r = 1)$, $(x = y)$, $(x < y)$, $(y = 2)$

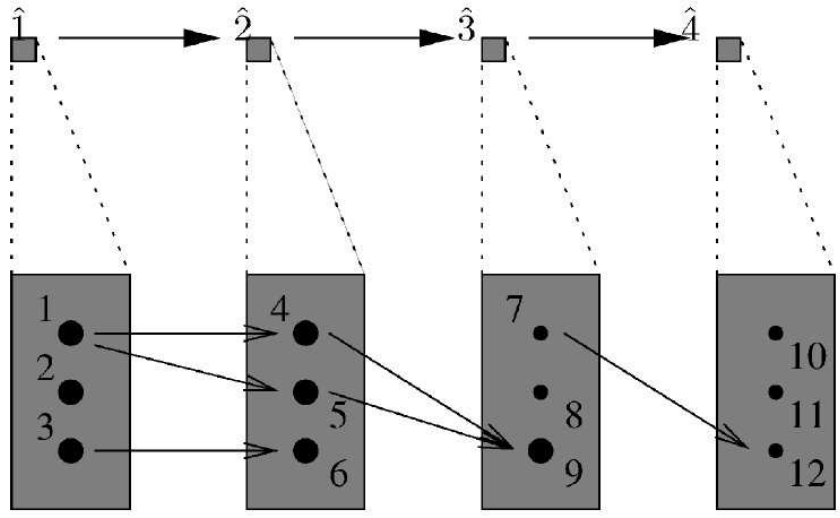
predykaty ze specyfikacji

$$h(r, x, y) = \langle (r = 1), (x = y), (x < y), (y = 2) \rangle$$

$$\langle r, x, y \rangle \approx_h \langle r', x', y' \rangle \iff \begin{array}{l} (r = 1 \iff r' = 1) \wedge \\ (x = y \iff x' = y') \wedge \\ (x < y \iff x' < y') \wedge \\ (y = 2 \iff y' = 2) \end{array}$$

CEGAR: fałszywy kontrprzykład

kontrprzykład abstrakcyjny $\widehat{1}\,\widehat{2}\,\widehat{3}\,\widehat{4}$



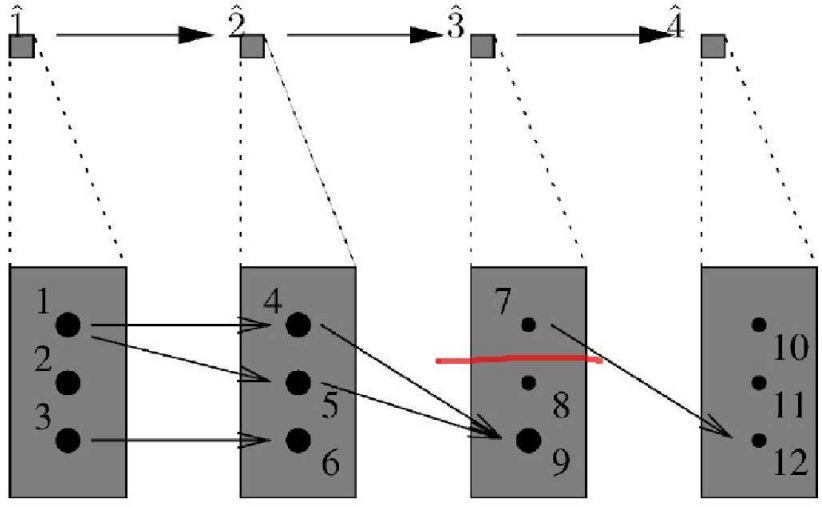
[Clarke, Grumberg, Jha, Lu, Veith 2003]

symulacja kontrprzykładu

$$S_1 := S_0 \cap h^{-1}(\,\widehat{1}\,) \qquad S_i := \vec{R}(S_{i-1}) \cap h^{-1}(\,\widehat{i}\,)$$

CEGAR: uszczegółowienie abstrakcji

kontrprzykład abstrakcyjny $\widehat{1}\,\widehat{2}\,\widehat{3}\,\widehat{4}$

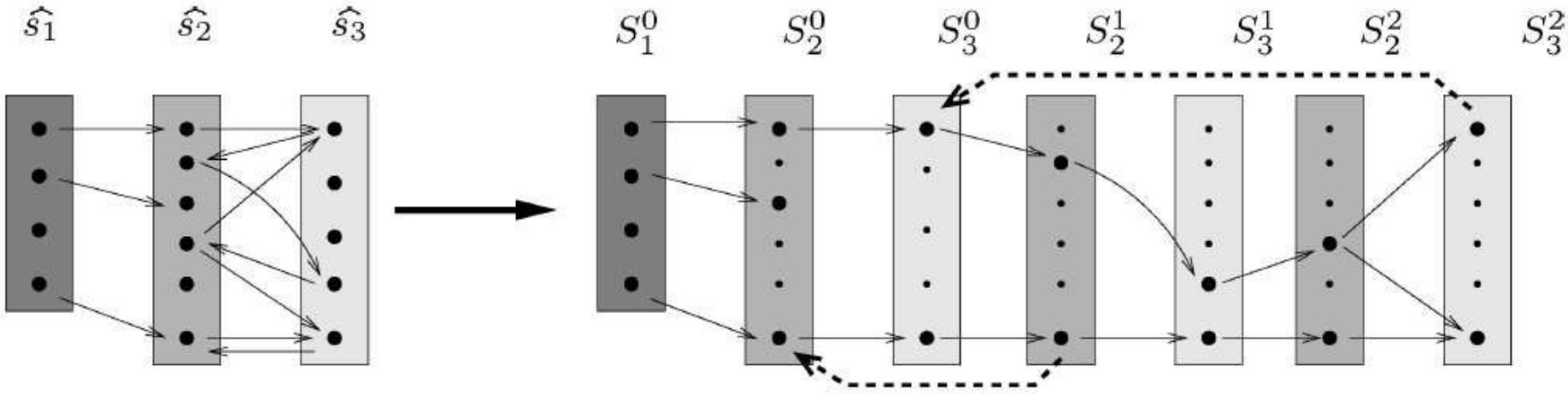


[Clarke, Grumberg, Jha, Lu, Veith 2003]

$$S_3 \cap \vec{R}^{-1}(h^{-1}(\widehat{4})) = \emptyset$$

CEGAR: pętla

kontrprzykład abstrakcyjny $\widehat{s_1}(\widehat{s_2}\widehat{s_3})^\omega$

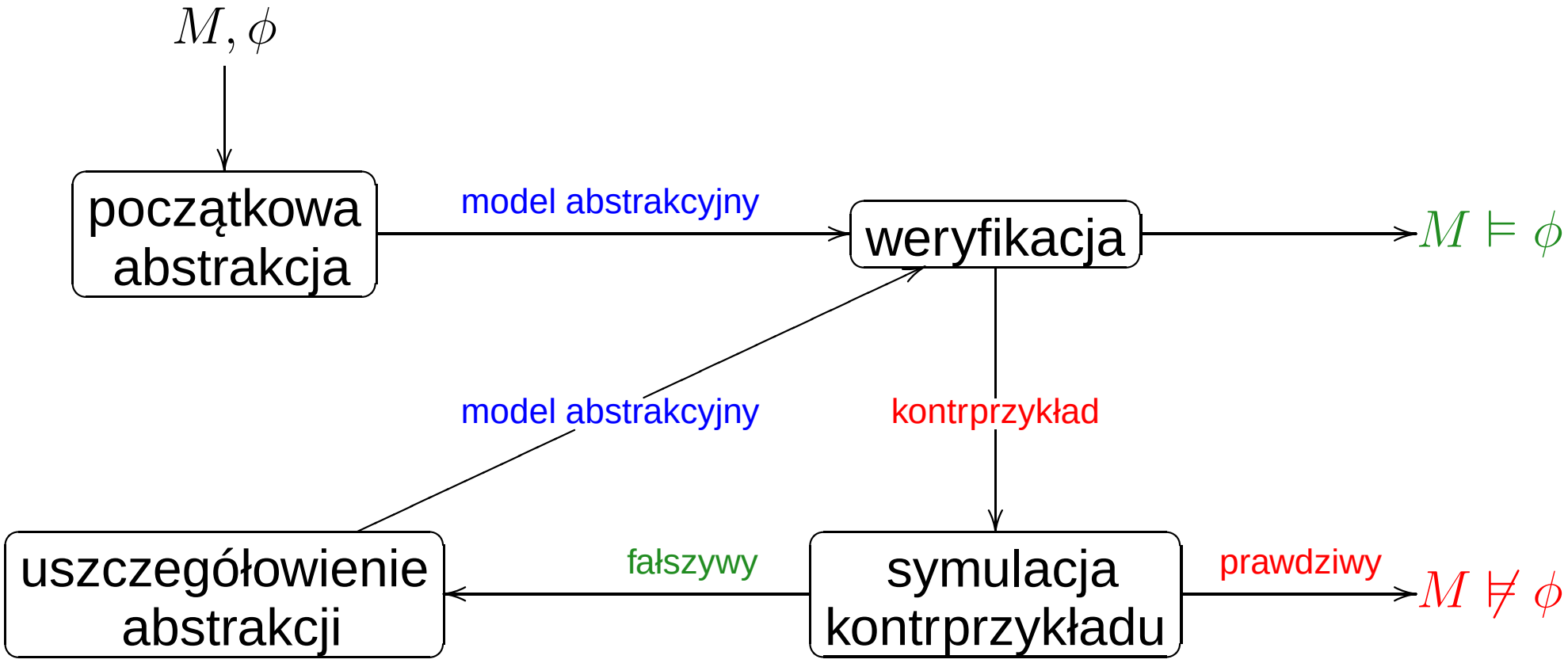


[Clarke, Grumberg, Jha, Lu, Veith 2003]

– rozwijamy pętlę $\min\{\text{rozmiar}(\widehat{s_2}), \text{rozmiar}(\widehat{s_3})\}$ razy

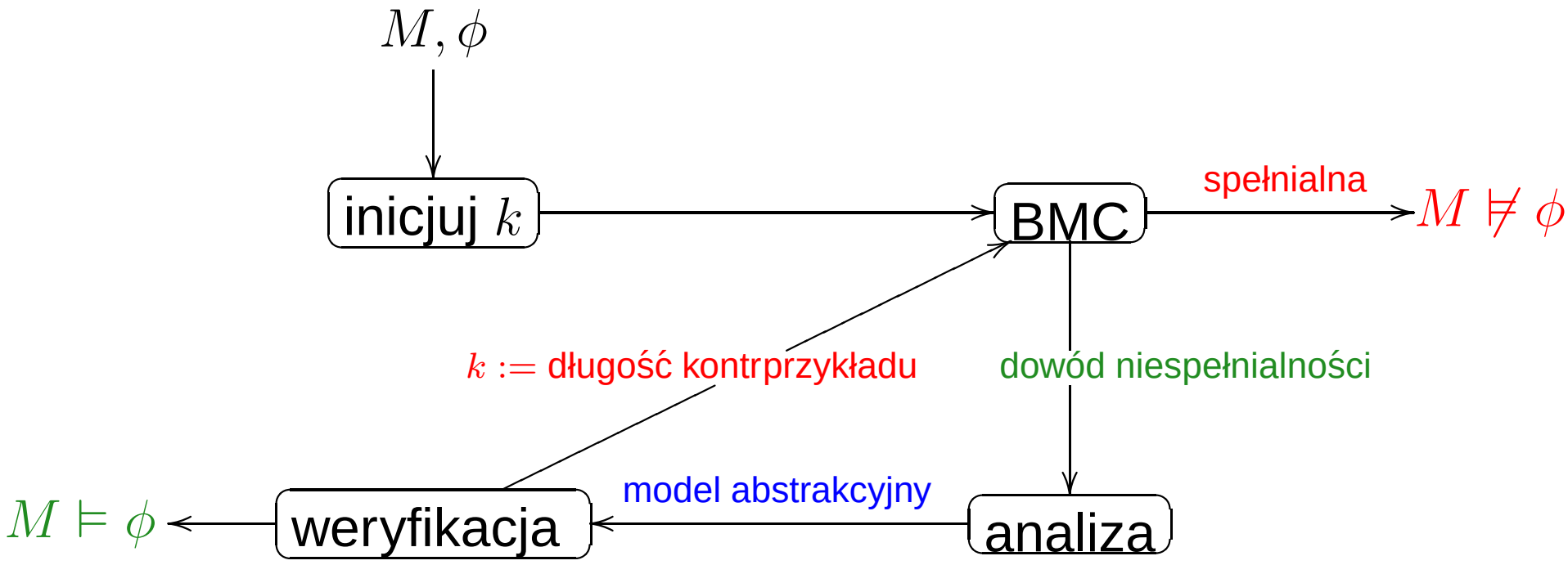
CEGAR: odmiany

– BDD + SAT: SAT-rozwiązywacz używany do symulacji kontrprzykładu i do uszczegółowienia abstrakcji



– BDD + SAT

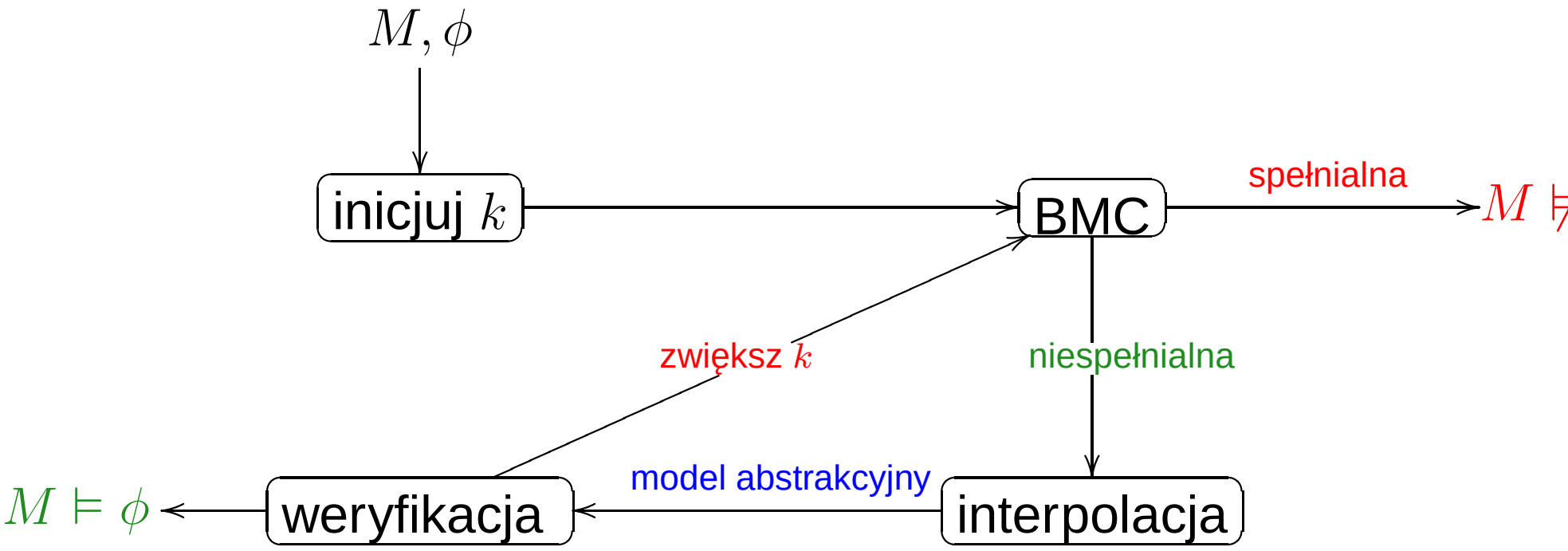
– abstrakcja z dowodu niespełnialności



odmiany (cd)

– BDD + SAT

– abstrakcja z interpolacji

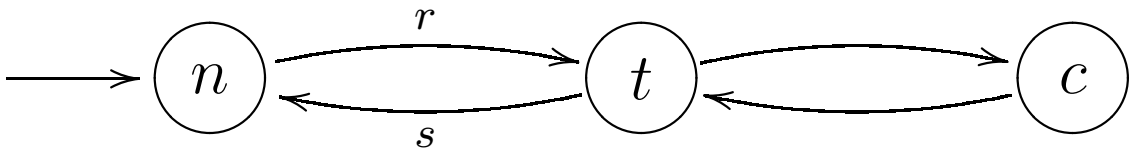


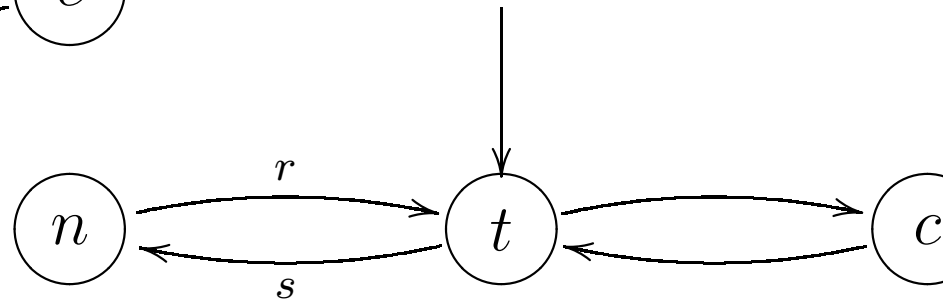
Podsumowanie

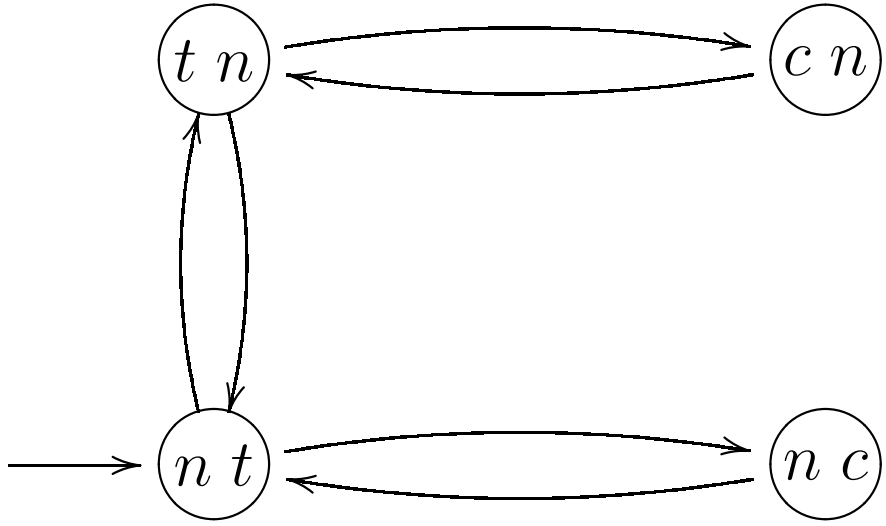
$M \sim M'$	$\forall \phi \in \mathrm{CTL}^*.\ M \models \phi \iff M' \models \phi$
$M \sim_F M'$	$\forall \phi \in \mathrm{CTL}^*.\ M \models_F \phi \iff M' \models_F \phi$
$M \preceq M'$	$\forall \phi \in \mathrm{ACTL}^*.\ M' \models \phi \implies M \models \phi$
$M \preceq_F M'$	$\forall \phi \in \mathrm{ACTL}^*.\ M' \models_F \phi \implies M \models_F \phi$
$M \simeq M'$	$\forall \phi \in \mathrm{ACTL}^*.\ M \models \phi \iff M' \models \phi$
$M \simeq_F M'$	$\forall \phi \in \mathrm{ACTL}^*.\ M \models_F \phi \iff M' \models_F \phi$
$L_\omega(M) \subseteq L_\omega(M')$	$\forall \phi \in \mathrm{LTL}.\ M' \models \phi \implies M \models \phi$
$L_\omega(M) = L_\omega(M')$	$\forall \phi \in \mathrm{LTL}.\ M \models \phi \iff M' \models \phi$

IV. Symetria

Symetria – przykład







– $V = \{x_1, \ldots, x_n\}, \qquad S = D^n$

– permutacja $\sigma : \{1 \ldots n\} \to \{1 \ldots n\}$ wyznacza

permutację $\rho : D^n \to D^n$:

$$\rho(d_1, \ldots, d_n) \;=\; (d_{\sigma(1)}, \ldots, d_{\sigma(n)})$$

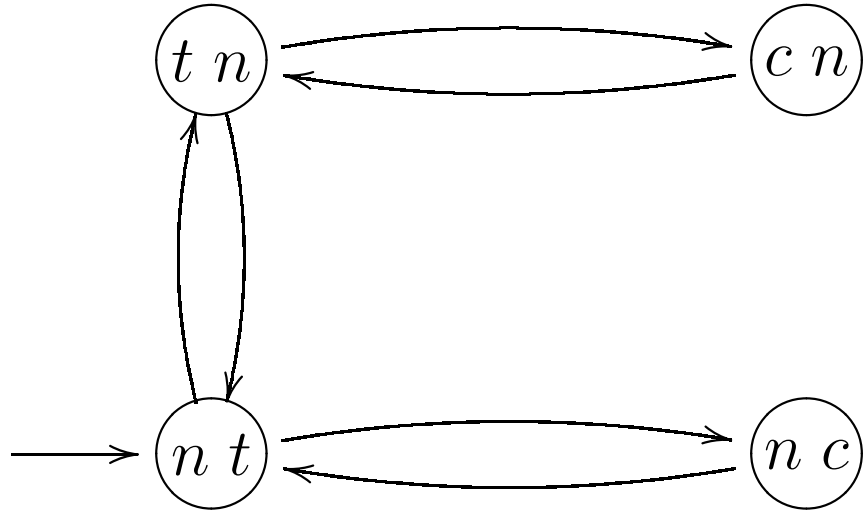
– grupa permutacji na $\{1 \ldots n\}$ (strukturze M) wyznacza

grupę permutacji G na D^n (przestrzeni stanów M)

Przykład: $D = \{n, t, c\}$

na $\{1, 2, \ldots, n\}$: grupa generowana przez $\sigma = (1\ 2 \ldots n)$

na D^n: grupa G generowana przez $\rho : \langle d_1 \ldots d_n \rangle \mapsto \langle d_2 \ldots d_n d_1 \rangle$



Model ilorazowy

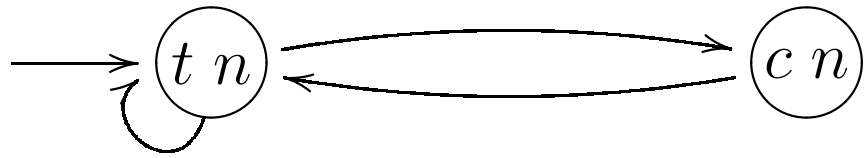
– G – grupa symetrii (automorfizmów) $\rho : D^n \to D^n$

$$R(\vec{d}, \vec{d'}) \implies R(\rho(\vec{d}), \rho(\vec{d'}))$$

– orbita $\theta_G(\vec{d})$ = $\{\vec{d'} \mid \exists \rho \in G.\ \rho(\vec{d}) = \vec{d'}\}$

– model ilorazowy $M/\approx_G$

$$\vec{d} \approx_G \vec{d'} \iff \vec{d} \in \theta_G(\vec{d'})$$



$$\phi \equiv (c_1 \implies \neg c_2) \wedge (c_2 \implies \neg c_1)$$

jest niezmiennicza względem G:

$$\vec{d} \vDash \phi,\ \rho \in G \implies \rho(\vec{d}) \vDash \phi$$

Tw.: $M \vDash \phi \iff M/\approx_G \vDash \phi$,

o ile ϕ jest niezmiennicza względem G

Tw.: $M \sim M/\approx_G$,

o ile wszystkie $p \in P$ są niezmiennicze względem G

Typowe przypadki:

Pierścień:

grupa generowana przez $\sigma = (1\ 2 \ldots n)$

Brak struktury:

grupa wszystkich permutacji

obliczamy $\approx_G$ jako najmniejszy punkt stały:

$$\vec{x} \approx_G \vec{z} \;\equiv\; \vec{x} = \vec{z} \vee \exists \vec{y}.\; \vec{x} \approx_G \vec{y} \wedge \bigvee_i \vec{z} = g_i(\vec{y})$$

$$R_G(\vec{x}, \vec{x'}) \;\equiv\; \exists \vec{y}, \vec{y'}.\, R(\vec{y}, \vec{y'}) \;\wedge\; \vec{y} \approx_G \vec{x} \wedge \vec{y'} \approx_G \vec{x'}$$

$\xi(\vec{d})$ – reprezentant klasy abs. $[\vec{d}]$

obliczamy relację $\vec{y} = \xi(\vec{x})$

$$R_G(\vec{x}, \vec{x'}) \;\equiv\; \exists \vec{y}, \vec{y'}.\, R(\vec{y}, \vec{y'}) \;\wedge\; \vec{y}{=}\xi(\vec{x}) \wedge \vec{y'}{=}\xi(\vec{x'})$$