Wyprowadzanie (uzasadnianie) tautologii metodami dedukcyjnymi wymaga znajomości pewnych podstawowych dla logiki praw logicznych. Są one wymienione w pliku:

http://partyka.pwsz.chelm.pl/PLS/
Wst-do-log-/Wst-do-log-_2020-10-20a.pdf

W przypadku trudności w znalezieniu znanych tautologii przy stosowaniu metod dedukcyjnych możemy posłużyć się techniką sprowadzania dowolnej formuły logicznej, ~~której~~ do tzw. postaci znormalizowanej formuły logicznej, której tautologiczność jest łatwa do rozstrzygnięcia metodami zero-jedynkowymi.

18.11.08

6. Postacie znormalizowane formuł logicznych

Def. 6.1

Dla dowolnego F: Var, F nazywamy

formułę zdaniową w postaci alternatywno-negacyjnej $:\Leftrightarrow F : \underline{ANForm}$, gdzie $\underline{ANForm}$ jest klasą formuł logicznych utworzonych w wyniku stosowania następujących reguł:

(6.1a):= $[p : \underline{Var} \,\vdots\, p : \underline{Char} \mapsto p : \underline{ANForm}] : \underline{RuL};$

(6.1b):= $[p : \underline{Var} \,\vdots\, p : \underline{Char} \mapsto \sim p : \underline{ANForm}] : \underline{RuL};$

(6.1c):= $[F, G : \underline{Var} \,\vdots\, F, G : \underline{ANForm} \mapsto (F) \vee (G) : \underline{ANForm}] : \underline{RuL};$

Def. 6.2

Dla dowolnego $F : \underline{Var}$, F nazywamy formułę zdaniową w postaci koniunkcyjno-negacyjnej $:\Leftrightarrow F : \underline{CNForm}$, gdzie $\underline{CNForm}$ jest klasą formuł logicznych utworzonych w wyniku stosowania następujących reguł:

(6.2a):= $[p : \underline{Var} \,\vdots\, p : \underline{Char} \mapsto p : \underline{CNForm}] : \underline{RuL};$

(6.2b):= $[p : \underline{Var} \,\vdots\, p : \underline{Char} \mapsto \sim p : \underline{CNForm}] : \underline{RuL};$

(6.2c):= $[F, G : \underline{Var} \,\vdots\, F, G : \underline{CNForm} \mapsto (F) \wedge (G) : \underline{CNForm}] : \underline{RuL};$

Def. 6.3

Dla dowolnego $F : \underline{Var}$, F nazywamy formułę zdaniową w postaci koniunkcyjno-alternatywno-negacyjnej $:\Leftrightarrow F : \underline{CANForm}$, gdzie $\underline{CANForm}$ jest klasą formuł logicznych utworzonych w wyniku stosowania następujących reguł:

(6.3a):= $[F : \underline{Var} \,\vdots\, F : \underline{ANForm} \mapsto F : \underline{CANForm}] : \underline{RuL};$

(6.3b):= $[F, G : \underline{Var} \,\vdots\, F, G : \underline{CANForm} \mapsto (F) \wedge (G) : \underline{CANForm}] : \underline{RuL};$

Def. 6.4

Dla dowolnego $F : \underline{Var}$, nazywamy formułę zdaniową w postaci alternatywno-koniunkcyjno-negacyjnej $:\Leftrightarrow F : \underline{ACNForm}$, gdzie $\underline{ACNForm}$ jest klasą formuł logicznych utworzonych w wyniku stosowania następujących reguł:

(6.4a):= $[F : \underline{Var} \,\vdots\, F : \underline{CNForm} \mapsto F : \underline{ACNForm}] : \underline{RuL};$

(6.4b):= $[F, G : \underline{Var} \,\vdots\, F, G : \underline{ACNForm} \mapsto (F) \vee (G) : \underline{ACNForm}] : \underline{RuL};$

Przykład 6.5

$p \lor q \lor \sim r \lor s$: ANForm $\quad \sim p \land \sim q \land r$: CNForm

(6.1a) (6.1a) (6.1b) (6.1a) (6.1a) (6.2b) (6.2a) (6.2c) (6.2a) (6.2c) (6.2b) (6.…)

(6.1c) (6.1c) (6.1c)

alternatywno-negacyjna

$\sim p \lor q \lor s$: ANForm

koniunkcyjno-negacyjna

$p \land r \land \sim s$: CNForm

$\underbrace{(p \lor q \lor \sim r \lor s)}_{(6.3a)} \overset{(6.3b)}{\land} \underbrace{(\sim p \lor q \lor s)}_{(6.3a)}$: CANForm

$\underbrace{(\sim p \land \sim q \land r)}_{(6.4a)} \overset{(6.4b)}{\lor} \underbrace{(p \land r \land \sim s)}_{(6.4a)}$: ACNForm

$(\sim p \land \sim q \land r) \lor (p \land r \land \sim s) \lor (\sim p \lor q \lor s)$: ACNForm

$(p \lor q \lor \sim r \lor s) \land (\sim p \lor q \lor s) \land p \land r \land \sim s$: CANForm

Natomiast formuły $\sim(p \lor q) \land r$, $p \land \sim(q \land r)$ nie są w żadnej z postaci określonych w def. 6.1–6.4.

Stosując prawa de Morgana możemy pierwszą formułę sprowadzić do formuły $\sim p \land \sim q \land r$: CNForm, która jest z nią logicznie równoważna.

Stosując prawo de Morgana do drugiej formuły otrzymujemy $p \land \sim(q \land r) \Leftrightarrow p \land (\sim q \lor \sim r)$, a więc $p \land (\sim q \lor \sim r)$: CANForm jest formułą logicznie równoważną z formułą $p \land \sim(q \land r)$.

Stosując następnie prawo rozdzielności koniunkcji względem alternatywy dostajemy $p \land (\sim q \lor \sim r) \Leftrightarrow (p \land \sim q) \lor (p \land \sim r)$, a więc $(p \land \sim q) \lor (p \land \sim r)$: ACNForm jest formułą równoważną z formułą $p \wedge \sim(q \wedge r)$

Def. 6.5

Dla dowolnego F: Var, F nazywamy formułę logiczną w postaci znormalizowanej $:\Leftrightarrow F$: CANForm lub F: ACNForm.

Def. 6.6

Dla dowolnego F: Var, F nazywamy formułę logiczną w postaci normalnej
$:\Leftrightarrow$ F:ACNForm

Uwaga 6.7

Stosując podstawowe prawa logiczne oraz regułę przechodności (tranzytywności) dla równoważności.

(6.7a):= $[F, G, H$: Var: $F \Leftrightarrow G$: Taut, $G \Leftrightarrow H$: Taut. $\vdash F \Leftrightarrow H$: Taut$]$: Rul;

Można dowolną formułę zdaniową F sprowadzić do logicznie równoważnej formuły F^*: ACNForm oraz F^{**}: CANForm.

$$a \cdot (b+c) = a \cdot b + a \cdot c$$

$$-(a+c) = (-a) + (-c)$$

Rozważmy formułę logiczną $F := (p \Rightarrow q) \wedge p \Rightarrow q$. Możemy formułę F sprowadzić do postaci normalnej w następujący sposób:

K1(:= $F \Leftrightarrow [(\sim p \vee q) \wedge p \Rightarrow q]$: Taut $\leftarrow \Vert$

RDS $\Vdash (\alpha \Rightarrow \beta) \Leftrightarrow (\sim \alpha \vee \beta)$;

K2(:= $F \Leftrightarrow \sim[(\sim p \vee q) \wedge p] \vee q$: Taut $\leftarrow \Vert$

RDS, (6.7a) $\Vdash (\alpha \Rightarrow \beta) \Leftrightarrow (\sim \alpha \vee \beta)$;

K3(:= $F \Leftrightarrow \sim(\sim p \vee q) \vee \sim p \vee q$: Taut $\leftarrow \Vert$

RDS, (6.7a) $\Vdash \sim(\alpha \wedge \beta) \Leftrightarrow \sim \alpha \vee \sim \beta$

K4(:= $F \Leftrightarrow (\sim\sim p \wedge \sim q) \vee \sim p \vee q$: Taut $\leftarrow \Vert$

RDS, (6.7a) $\Vdash \sim(\alpha \vee \beta) \Leftrightarrow \sim \alpha \wedge \sim \beta$

K5(:= $F \Leftrightarrow (p \wedge \sim q) \vee \sim p \vee q$: Taut $\leftarrow \Vert$

RDS, (6.7a) $\Vdash \sim\sim \alpha \Leftrightarrow \alpha$

K6(:= $F \Leftrightarrow (p \vee \sim p \vee q) \wedge (\sim q \vee \sim p \vee q)$: Taut $\leftarrow \Vert$

RDS, (6.7a) $\Vdash (\alpha \wedge \beta) \vee \gamma \Leftrightarrow (\alpha \vee \gamma) \wedge (\beta \vee \gamma)$;

Przyjmując $F^* := (p \wedge \sim q) \vee \sim p \vee q$
i $F^{**} := (p \vee \sim p \vee q) \wedge (\sim q \vee \sim p \vee q)$
widzimy, że
$F \Leftrightarrow F^*$ i $F \Leftrightarrow F^{**}$ oraz F^*: ACNForm
i F^{**}: CANForm.

Symbolem RDS oznaczamy tutaj zespół reguł równoważnego dołączania spójników logicznych. Dla przykładu reguła dołączania koniunkcji wygląda następująco:

(6.7b):= $[F, G, H : Var : F \Leftrightarrow G : Taut,\ H : WForm \vdash\!> F \wedge H \Leftrightarrow G \wedge H : Taut] : Rul;$

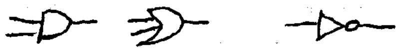
Pozostałe reguły równoważnego dołączania spójników logicznych powstają przez zastąpienie w regule (6.7b) koniunkcji przez odpowiednie spójniki logiczne. Dodatkowo uwzględniamy regułę obustronnego przeczenia równoważności:

(6.7c):= $[F, G : Var : F \Leftrightarrow G : Taut \vdash\!> \sim F \Leftrightarrow \sim G : Taut] : Rul;$

25.11.08

Uwaga 6.8

Fakt, że dowolną formułę logiczną F: WForm można wyrazić przez równoważną jej formułę F^*: ACNForm ma duże znaczenie w logice i jej zastosowaniach. Dzięki temu dowolny warunek logiczny w językach programowania można wyrazić wyłącznie w terminach operatorów logicznych: koniunkcji „and", alternatywy „or" i negacji „not". W konsekwencji cyfrowy układ logiczny realizujący wartościowanie dowolnie zadanej funkcji zdaniowej (dowolnie zadany warunek logiczny) można zrealizować za pomocą podstawowych cyfrowych układów logicznych (tzn. bramek logicznych) realizujących wartościowanie operatorów logicznych: koniunkcji, alternatywy i negacji.



Uwaga 6.9
Metodę sprowadzania dowolnej formuły logicznej F: WForm do równoważnej z nią postaci F^*: CANForm można wykorzystać do dowodu tautologiczności formuły F, bądź jej nietautologiczności. Na podstawie reguł tworzenia formuł z klasy CANForm stwierdzamy, że
$F^* = F_1 \wedge F_2 \wedge \ldots \wedge F_m$ dla pewnych formuł $F_1, F_2, \ldots, F_m$ w postaci alternatywno-negacyjnej: ANForm.
Korzystając z reguły
(6.9a):= $[P, Q$: Var; $P \wedge Q$: Taut ⇸ P, Q : Taut$]$: Rul;

widzimy, że formuła F^*: Taut wtedy i tylko wtedy, gdy wszystkie formuły alternatywno-negacyjne $F_1, F_2, \ldots, F_m$ wchodzące w skład formuły F^* są tautologiami. Zatem problem tautologiczności formuły F^* sprowadza się do problemu badania tautologiczności dowolnie zadanej formuły G: ANForm. Na podstawie reguł tworzenia formuł z klasy ANForm oraz reguły wartościowania alternatywy stwierdzamy, że G: Taut wtedy i tylko wtedy, gdy G zawiera parę formuł logicznych postaci $p, \sim p$ dla pewnego p: Char. Jeśli zatem uda nam się stwierdzić na podstawie powyższych rozważań, że F^*: Taut, to korzystając z faktu, że $F \Leftrightarrow F^*$: Taut stwierdzamy na podstawie reguły tranzytywności (6.7b), że F: Taut. Jeśli zaś F* nie będzie

tautologią, to F również nie nie

będzie. Dla przykładu formuła

logiczna F = (p => q) ∧ p => q

rozważona w uwadze 6.7 da się

sprowadzić do postaci

F** = (p ∨ ~p ∨ q) ∧ (~q ∨ ~p ∨ q): CANFor

Ponieważ F**: Taut i F <=> F**: Taut

a więc F: Taut.