Tentamen i TDDC75 Diskreta strukturer

2016-10-27: Lösningsförslag

1. (a) Sanningstabellen för uttrycket är:

q	p	$\neg q$	$\neg p$	$(\neg p \vee \neg q)$	$(p \vee q)$	$((p \vee q) \wedge (\neg p \vee \neg q))$
0	0	1	1	1	0	0
0	1	1	0	1	1	1
1	0	0	1	1	1	1
1	1	0	0	0	1	0

   (b) Bevis för $(p \vee q) \wedge (\neg p \vee \neg q) \models \neg p \leftrightarrow q$

(1)	$(p \vee q) \wedge (\neg p \vee \neg q)$	(Premiss)
(2)	$(\neg\neg p \vee q) \wedge (\neg p \vee \neg q)$	(1) och dubbel negation
(3)	$(\neg p \rightarrow q) \wedge (\neg p \vee \neg q)$	(2) och implikationslagen
(4)	$(\neg p \rightarrow q) \wedge (q \rightarrow \neg p)$	(3) och implikationslagen
(5)	$(\neg p \leftrightarrow q)$	(4) och ekvivalenslagen

   (c) Vi kan visa detta genom att sätta upp sanningstabellen för $\neg p \leftrightarrow q$ och konstatera att uttrycket i alla tolkningar är samma som för $(p \vee q) \wedge (\neg p \vee \neg q)$.

q	p	$\neg p$	$(\neg p \leftrightarrow q)$
0	0	1	0
0	1	0	1
1	0	1	1
1	1	0	0

2. (a) Falskt. $\{f_1\}$ är en mängd och inte en fil så $\{f_1\} \notin F$.

   (b) Sant. För det enda element i $\{f_1\}$ gäller $f_1 \in F$, så $\{f_1\} \subseteq F$. Vidare gäller inte $F \subseteq \{f_1\}$ eftersom $f_2 \in F$, men $f_2 \notin \{f_1\}$. Alltså gäller $\{f_1\} \subset F$.

   (c) Falskt. $F \cap \{f_3\} = \{f_3\}$ och eftersom $f_1 \in \{f_1, f_2\}$ men $f_1 \notin \{f_3\}$ så stämmer inte påståendet.

   (d) Falskt. Alla element i 2^F är mängder. Alltså gäller till exempel inte att $f_1 \in 2^F$.

   (e) Tillräcklig iformation saknas. $|2^A| = 2^{|A|} = 2^2 = 4$ och $|2^B| = 2^{|B|}$ så för att besvara frågan måste vi undersöka kardinaliteten av $B = F \setminus \{f_1, f_2\}$. Vi vet inte hur många filer som finns på hårddisken, bara att $\{f_1, f_2, f_3\} \subseteq F$ så att $B = F \setminus \{f_1, f_2\}$ innehåller minst 1 element, så $2^{|B|} \geq 2$.

3. Vi visar med induktion. Låt $P(n)$ vara påståendet att R^n är symmetrisk om R är symmetrisk.

   - Bassteget: Vi visar att $P(0)$ gäller, dvs att R^0 är symmetrisk. Av definitionen följer att $R^0 = id_A$ vilken är symmetrisk eftersom $id_A = \{(a, a) \mid a \in A\}$.
   - Induktionssteget: Anta $P(k)$ för något $k \geq 0$. Vi ska då visa att $P(k+1)$ gäller, dvs att R^{k+1} är symmetrisk. Om $R^{k+1} = \emptyset$ gäller $P(k+1)$ trivialt, så vi antar att det finns minst ett par i denna relation. Betrakta då godtyckligt par $(a, c) \in R^{k+1} = R \circ R^k$. Då gäller att det finns ett element $b \in A$ sådant att $(a, b) \in R^k$ och $(b, c) \in R$. Av induktionsantagandet vet vi att även $(b, a) \in R^k$ premissen ger att R är symmetrisk så $(c, b) \in R$. Eftersom $(b, a) \in R^k$ och $(c, b) \in R$ gäller att $(c, a) \in R^k \circ R = R \circ R^k = R^{k+1}$. $R^k \circ R = R \circ R^k$ gäller pga att sammansättning av relationer är associativt. Vi utgick ifrån ett godtyckligt par $(a, c) \in R^{k+1}$ och kunde finna $(c, a) \in R^{k+1}$, alltså gäller detta för alla par i relationen, vilken således är symmetrisk.

   Av bassteget, induktionssteget och induktionsprincipen följer $P(n)$ för alla $n \geq 0$, så R^n är symmetrisk för alla $n \geq 0$ om R är symmetrisk.

4. (a) $d_K(b) = e_K^{-1}(b)$ där $b \in B$ och e_K^{-1} betecknar inversen till krypteringsfunktionen.

   (b) e_K är alltid injektiv eftersom den har en invers. Om det fanns två meddelanden $m_1, m_2 \in A, m_1 \neq m_2$ sådana att $e_K(m_1) = e_K(m_2)$ så skulle det inte kunna finnas en dekrypteringsfunktion d_K som återställer båda till klartext.

   (c) Elementen i B måste vara minst 128 bitar eftersom det faktum att e_K är injektiv implicerar att $|A| \leq |B|$. Om e_K även är surjektiv så är e_K en bijektion så $|A| = |B|$ och elementen i B måste vara exakt 128 bitar långa. Om e_K inte är surjektiv kan de krypterade meddelandena vara längre.

5. Anta motsatsen, att $B \neq C$. Då måste det finnas minst ett element x som finns i någon av mängderna, men inte den andra. Anta utan begräninsning att $x \in B$ och $x \notin C$. Då gäller att $x \in A \cup B = A \cup C$, dvs $x \in A$ eller $x \in C$. Men eftersom $x \notin C$ så följer att $x \in A$, och därmed $x \notin \overline{A}$.

   På samma sätt kan vi konstatera att av $x \in B$ följer att $x \in \overline{A} \cup B = \overline{A} \cup C$. Men $x \notin C$ och $x \notin \overline{A}$ vilket motsäger $x \in \overline{A} \cup C$ och vårt antagande $B \neq C$ är därmed falskt.

   Av ovanstående följer att $B = C$.

6. (a) Ibland. Betrakta följande två program:

P1:
(1) $a = 1$
(2) $b = a$

P2:
(1) $a = 1$
(2) $b = a$
(3) $a = b$

För P1 har vi $T_1 = \emptyset$, $T_2 = \{(b, a)\}$, $R_0 = R_1 = \{(a, a), (b, b)\}$, $R = R_2 = \{(b, a), (a, a), (b, b)\}$. R är en partialordning eftersom den är reflexiv, transitiv och anti-symmetrisk.

För P2 har vi $T_1 = \emptyset$, $T_2 = \{(b, a)\}$, $T_3 = \{(a, b)\}$, $R_0 = R_1 = \{(a, a), (b, b)\}$, $R_2 = \{(b, a), (a, a), (b, b)\}$ och $R = R_3 = \{(a, b), (b, a), (a, a), (b, b)\}$. R är då inte en partialordning eftersom den inte är anti-symmetrisk.

(b) Påståendet säger att om (v_d, v) är med i R så är $v_d = v$ dvs den enda variabel som påverkas av v är v självt. Alltså finns det ingen tilldelning i programmet där v förekommer på högersidan.

(c) Vi kan först konstatera att E är en ekvivalensrelation över alla variabler.

- Reflexiv eftesom R är reflexiv (egenskapen bevaras under symmetrisk och transitivt hölje).
- Symmetrisk eftersom den är skapad som det transitiva höljet av en symmetrisk relation (se uppgift 3).
- Transitiv eftersom den är definerad som ett det transitiva höljet av en annan relation.

Alla variabler som på något sätt kan påverka varandra hamnar i samma ekvivalensklass. Om två variabler $(v_1, v_2) \notin E$ innebär det att de är helt oberoende av varandra. Således kan programrader som tilldelar värden till dessa två variabler köras oberoende och i godtycklig inbördes ordning (obs att eventuella variabler på högersidan med nödvändighet måste vara i samma ekvivalensklass).

Utökad kommentar (ej förväntat svar): Om en dator har flera kärnor eller processorer, så kan programmet delas upp i flera oberoende delar som körs parallellt.

I verkligheten finns fler faktorer att ta hänsyn till och mer avancerade metoder för att lösa detta problem. Risken med ovanstående metod är att alla variabler hamnar i samma ekvivalensklass.