UNIVERSITETET I OSLO

Det matematisk-naturvitenskapelige fakultet

Eksamen i	INF2080 — Logikk og beregninger
Eksamensdag:	3. juni 2013
Tid for eksamen:	14.30 – 18.30
Oppgavesettet er på 3 sider.	
Vedlegg:	Ingen
Tillatte hjelpemidler:	Ingen

Kontroller at oppgavesettet er komplett før
du begynner å besvare spørsmålene.

Gjør dine egne forutsettninger dersom du er i tvil om hvordan oppgaveteksten skal tolkes.

DEL I (Automater, språk og beregnbarhetsteori)

Oppgave 1

- Gi en DFA som gjenkjenner språket gitt ved det regulære uttrykket $a(b^*)$.
- Gi en DFA som gjenkjenner språket gitt ved det regulære uttrykket $a(c \cup (b^*))$.
- Forenkl det regulære uttrykket $((a^*)(b^*))^* \cup (a^*)$. (Du skal altså finne et enklere regulært uttrykk som beskriver samme språk.)

(Slutt på oppgaven.)

Språkene A og B er gitt ved

$$A = \{s \mid s \in \{a,b\}^* \text{ og antall } a\text{'er i } s \text{ er delelig med 2}\}$$

og

$$B = \{s \mid s \in \{a,b\}^* \text{ og antall } a\text{'er i } s \text{ er delelig med 3}\}$$

Oppgave 2

- Gi en DFA som aksepterer språket A.
- Gi en DFA som aksepterer språket B.
- Gi en NFA som aksepterer språket $A \cup B$.

(Fortsettes på side 2.)

(Slutt på oppgaven.)

La Σ være et alfabet som består av to parantestegn $()$ – en venstreparantes og en høyreparantes. La C være det språket over Σ som består av alle strenger Σ^* der parantessettingen er balansert på vanlig måte. Så $()(()())$ er et eksempel på en streng som er i språket C, mens $)((()((((()$ er et eksempel på en streng som ikke er i språket C.

Oppgave 3

- Gi en PDA som gjenkjenner språket C.
- Gi en CFG som genererer språket C.

Oppgave 4

Gi et språk som ikke kan gjenkjennes av en PDA, men som kan gjenkjennes av en turingmaskin. Forklar hvorfor dette språket ikke kan gjenkjennes av en PDA. Og forklar hvorfor det kan gjenkjennes av en turingmaskin.

DEL II (Kompleksitetsteori)

Språket *PATH* er mengden

$$\{\langle G, s, t\rangle \mid G \text{ er en rettet graf hvor det går en rettet sti fra } s \text{ til } t\} .$$

Språket er kjent fra læreboken. De to teoremene nedenfor er bevist i læreboken.

Teorem 1. *PATH* er *NL*-komplett.

Teorem 2. *PATH* $\in$ *P*.

Oppgave 5

Forklar hva det betyr at et problem er *NL*-komplett. Svar kort.

Oppgave 6

Gi en kort skisse av et bevis for Teorem 2. *(Slutt på oppgaven.)*

La σ være en sti i en graf G, og la $a_1, \ldots, a_n$ være en sekvens med noder fra G. Vi sier at σ *følger* $a_1, \ldots, a_n$ dersom σ passerer nodene $a_1, \ldots, a_n$ i den nevnte rekkefølge. En sti som følger en sekvens, kan passere noder som ikke forekommer i sekvensen, men den skal passere alle nodene i sekvensen, og den skal passere den første noden i sekvensen før den passerer den andre noden i sekvensen, og den skal passere den andre

(Fortsettes på side 3.)

noden i sekvensen før den passerer den tredje noden i sekvensen, og så videre.

Språket *FPATH* er mengden

$$\{\langle G, a_1, \ldots, a_n\rangle \mid G \text{ er en rettet graf med en rettet sti som følger sekvensen } a_1, \ldots, a_n\}\,.$$

Oppgave 7

Vis at *FPATH* er *NL*-komplett. Du kan bruke Teorem 1. *(Du skal* ikke *vise Teorem 1. Slutt på oppgaven)*

La σ være en sti i en graf G, og la $a_1, \ldots, a_n$ være en sekvens med noder fra G. Vi sier at σ *dekker* sekvensen $a_1, \ldots, a_n$ dersom σ er innom hver av nodene i sekvensen $a_1, \ldots, a_n$ nøyaktig en gang. En sti som dekker en sekvens kan passere noder som ikke forekommer i sekvensen. En sti som dekker en sekvens skal passere alle nodene som forekommer i sekvensen en – og bare en – gang, med det er likegyldig i hvilken rekkefølge stien passerer disse nodene.

Språket *DPATH* er mengden

$$\{\langle G, a_1, \ldots, a_n\rangle \mid G \text{ er en rettet graf med en rettet sti som dekker sekvensen } a_1, \ldots, a_n\}\,.$$

Oppgave 8

- Er *DPATH* i *NL*?
- Er *DPATH* *NL*-komplett?
- Er *DPATH* i P?
- Er *DPATH* *NP*-komplett?
- Er *DPATH* i *PSPACE*?

Begrunn svarene. Du må gjerne svare under rimelige antagelser, som f.eks. at $P \neq NP$, men da bør du gjøre disse antagelsene eksplisitt.