

Programmering og Problemløsning

Datalogisk Institut, Københavns Universitet

Uge(r)seddel 8 - gruppeopgave

Jon Sparring

26. oktober - 24. november

I denne ugeseddels periode skal vi arbejde med 3 læringsmål:

1. Fsharp moduler
2. Ind- og udlæsning
3. Unit-testing
4. Rapportskrivning

Bemærk at perioden er lang, hvilket skyldes at der ingen undervisning (hverken forelæsninger eller øvelser) er i perioden 2.-15. november, hvilket skyldes skiftet mellem blok 1 og 2, og selvom dette er et semesterkursus, skal vi respektere undervisningspausen mellem blokkene. Helt konkret sidste uge i blokken dvs. 2.-8. november er en kursusuge, som oftest bruges til eksamen, og uanset betragtes som en uge, I skal studere i, mens mellemugen dvs. 9.-15. november er ikke en del af kurset. Opgavens længde er derfor planlagt til at svarer til ca. 3 almindelige ugesedler.

Afleveringsopgave

Temaet for uernes opgaver er at programmere et Sudoku-spil. *Sudoku* er et puslespil, som er blevet opfundet uafhængigt flere gange; den tidligste “ægte” version af sudoku synes at kunne spores tilbage til det franske dagblad *Le Siècle* i 1892.

Vi betragter her kun den basale variant, som spilles på en matrix af 81 små felter, arrangeret i 9 rækker og 9 søjler. Matricen er desuden inddelt i 9 “bokse” eller “regioner”, hver med 3 gange 3 felter.

Nogle af felterne er udfyldt på forhånd, og puslespillet går ud på at udfylde de resterende felter på en sådan måde, at hver af de 9 rækker, hver af de 9 søjler og hver af de 9 regioner

kommer til at indeholde en permutation af symbolerne fra et forelagt alfabet af størrelse 9; vi vælger her (som man plejer at se det) alfabetet bestående af cifrene fra 1 til 9.

Her er en lovlig starttilstand for et spil sudoku:

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Og følgende er en vindende tilstand (en "løsning") af ovenstående:

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Nummerering og filformat

Lad os nummerere rækkerne $r = 0, 1, \dots, 8$ ovenfra og ned og søjlerne $s = 0, 1, \dots, 8$ fra venstre mod højre. Også regionerne vil vi nummerere $q = 0, 1, \dots, 8$, i “normal læse-retning” (for vestlige sprog). Sammenhængen mellem rækkenummer r , søjlenummer s og regionsnummer q kunne så udtrykkes i følgende formel ved brug af heltalsoperationer:

- Feltet i række nummer r og søjle nummer s vil ligge i region nummer

$$q = r / 3 * 3 + s / 3 \quad (1)$$

- Region nummer q består af felterne

$$(r, s) = (q / 3 * 3 + m, q \% 3 * 3 + n); m, n \in \{0, 1, 2\}. \quad (2)$$

En spiltilstand kan gemmes i en fil, ved at antage, at indholdet *altid* ser ud som følger:

- Der er mindst 90 tegn i filen (der kan være flere, men vi er kun interesseret i de 90 første)
- De første 90 tegn i filen er delt op i 9 grupper, som repræsenterer indholdet af række 0, ..., 8 i nævnte rækkefølge. Hver gruppe består af 10 tegn: Først 9 tegn, som er et blandt '1', ..., '9', '*', og til sidst strengen "\n".

F.eks. er nedenstående indholdet i en fil, som indeholder starttilstanden for ovenstående sudoku:

```
53**7****\n6**195***\n*98****6*\n8***6***3\n4**8*3**1\n7***2***6\n*6****28*\n***419**5\n****8**79\n
```

Hvis vi fortolker strengen "\n" som “ny linje”, bliver ovenstående lettere at læse:

```
53**7****
6**195***
*98****6*
8***6***3
4**8*3**1
7***2***6
*6****28*
***419**5
****8**79

```

Krav til opgavebesvarelsen

I skal programmere et Sudoku spil og skrive en rapport. Afleveringen skal bestå af en pdf indeholdende rapporten, et katalog med et eller flere fsharp programmer som kan over-sættes med Monos fsharpc kommando og derefter køres i mono, og en tekstfil der angiver

sekvensen af oversættelseskommandoer nødvendigt for at oversætte jeres program(mer). Kataloget skal zippes og uploades som en enkelt fil. Kravene til programmeringsdelen er:

1. Programmet skal kunne læse en (start-)tilstand fra en fil.
2. Brugeren skal kunne indtaste filnavnet for (start-)tilstanden
3. Brugeren skal kunne indtaste triplen (r, s, v) , og hvis feltet er tomt og indtastningen overholder spillets regler, skal matrixen opdateres, og ellers skal der udskrives en fejlmeddelelse på skærmen
4. Programmet skal kunne skrive matrixens tilstand på skærmen (på en overskuelig måde)
5. Programmet skal kunne foreslå lovlige tripler (r, s, v) .
6. Programmet skal kunne afgøre, om spillet er slut.
7. Brugeren skal have mulighed for at afslutte spillet og gemme tilstanden i en fil.
8. Programmet skal kommenteres ved brug af fsharp kommentarstandarden
9. Programmet skal struktureres ved brug af et eller flere moduler, som I selv har skrevet
10. Programmet skal unit-testes

Kravene til rapporten er:

11. Rapporten skal skrives i L^AT_EX.
12. I skal bruge `rapport.tex` skabelonen
13. Rapporten skal som minimum i hoveddelen indeholde afsnittene Introduktion, Problemformulering, Problemanalyse og design, Programbeskrivelse, Afprøvning, og Diskussion og Konklusion. Som bilag skal I vedlægge afsnittene Brugervejledning og Programtekst.
14. Alle gruppemedlemmer skal give feedback på et af hovedafsnittene i en anden gruppes rapport. Hvis og hvilke dele I gav feedback og hvem der gav feedback på jeres rapport skal skrives i Forordet i rapporten.
15. Rapporten må maksimalt være på 20 sider alt inklusivt.

Bemærk, at Sudoku eksemplerne i denne tekst er sat med L^AT_EX-pakken `sudoku`.

Øvelsesopgaver

Til øvelserne på alm. skema forventer vi at I arbejder efter nedenstående plan.

Mandag-Tirsdag 26-27/10: Evaluering på udvalgte hold, HR 7.1, 7.4, 7.5, 7.7

Fredag 30/10: Feedback på 6g, HR: 7.8. Resterende tid arbejdes med afleveringsopgaven.

Mandag 16/11: Skriv unit-tests for udleveret program, som bliver offentliggjort inden mandag 16/11. Arbejd med unit-tests for afleveringsopgaven.

Tirsdag 17/11: Skriveworkshop - Arbejd på rapporten i afleveringsopgaven med fokus på Problemformulering, Problemanalyse og design, og Afprøvning

Fredag 20/11: Skriveworkshop - Gruppe-til-gruppe feedback

God fornøjelse.