



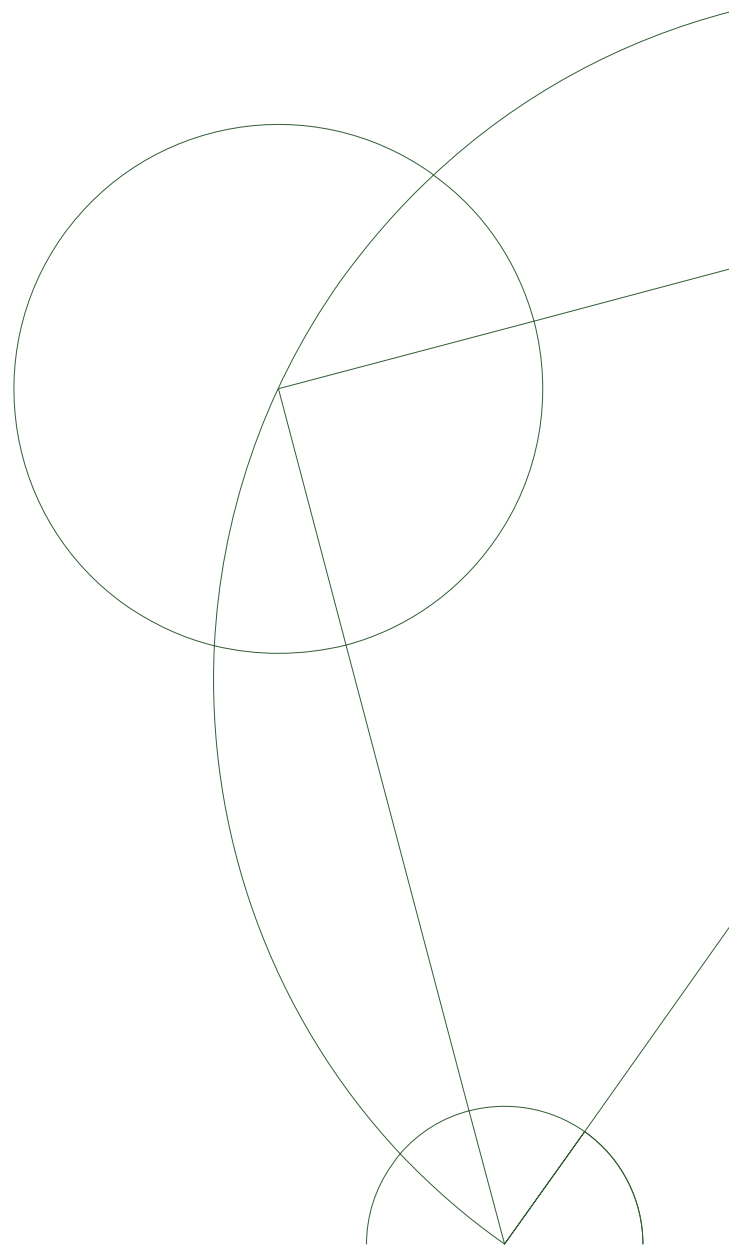
# Programmering og Problemløsning

## Aflevering 2i

Adam Frederik Ingwersen Linnemann,  
GQR701  
Hold 4

Datalogisk Institut  
Københavns Universitet

September 20, 2016



## 2i.0

Denne delopgave beskæftiger sig med metasyntaksen 'Extended Backus-Naur Form' (EBNF). Vi betragter her en EBNF der indeholder 4 tokens.

1. `charLiteral` kan antage enhver unicode værdi
2. `stringLiteral` kan antage enhver unicode værdi i citationstegn
3. `operator` kan udelukkende antage værdien `'+'`
4. `expression` kan enten antage værdi i form af `stringLiteral` eller `stringLiteral` efterfulgt af `operator` efterfulgt af `expression`

Expressions kan udtrykkes udelukkende af kombinationer af tokenener - som i denne sammenhæng ikke har noget eksplicit indhold:

1.  $\text{Exp1} \leftarrow \text{stringLiteral}$
2.  $\text{Exp2} \leftarrow \text{stringLiteral}, \text{operator}, \text{stringLiteral}, \text{operator}, \text{Exp1}$
3.  $\text{Exp3} \leftarrow \text{stringliteral}, \text{operator}, \text{Exp2}$   
 $= \text{stringliteral}, \text{operator}, \text{stringLiteral}, \text{operator}, \text{stringLiteral}, \text{operator}, \text{stringLiteral}$

Hver enkelt token er defineret ved en eller flere terminaler - som er antager værdier som f.eks. unicode karakterer. Terminaler kan sammensættes inden for reglementet defineret i hvert token. Givet betragtede EBNF har vi, at mulige kombinationer kan være et arbitrært antal adderede strenge. Af mulige kombinationer, er 3 anført i listen nedenfor:

1. `"Chika chika" → "Chika chika"`
2. `"What?"+"..."+"My name is..."+"Who?..."+[Expression 1] → "What?...My name is...Who?...Chika chika"`
3. `"Hi.. My name is.. " + [Expression 2] → "Hi.. My name is.. What?...My name is...Who?...Chika chika"`

EBNF'en er ikke i stand til at arbejde med værdier, som ikke er defineret i et token. Eksempler på situationer, hvor sekvenser er ikke-gyldige er angivet nedenfor:

1. `stringLiteral, stringLiteral, operator, operator`
2. `"Hej"*"Hej"`
3. `amamdmaudjwjadjaw`

## 2i.1

Decimal	Binær	Heximal	Oktal
10	01010	A	12
21	10101	15	25
63	00111111	3f	77
63	00111111	3f	77

## 2i.2

I denne delopgave, var jeg i tvivl om, hvad der bliver spurgt efter. Jeg har udarbejdet en besvarelse, som var i overensstemmelse med min umiddelbare opfattelse af spørgsmålet:

```
///    Mulighed 1)
let streng = "hello world"
printfn "%A" (streng.[0..4] + streng.[6..10])

///    Mulighed 2)
let subStreng1 = streng.[0..4]
let subStreng2 = streng.[6..10]
printfn "%A" subStreng1
printfn "%A" subStreng2
```