INF 3230: Videregående kurs i formell modellering

Peter Ølveczky

Universitetet i Oslo

9. mai 2012

Eksekverbar Formell Modellering

- Modellering: lage en modell av systemdesign før implementasjon
- Formell: mulig å utlede egenskaper/konsekvenser av modellen
- Eksekverbar: la datamaskinen analysere modellen

Eksekverbar Formell Modellering

- Modellering: lage en modell av systemdesign før implementasjon
- Formell: mulig å utlede egenskaper/konsekvenser av modellen
- Eksekverbar: la datamaskinen analysere modellen

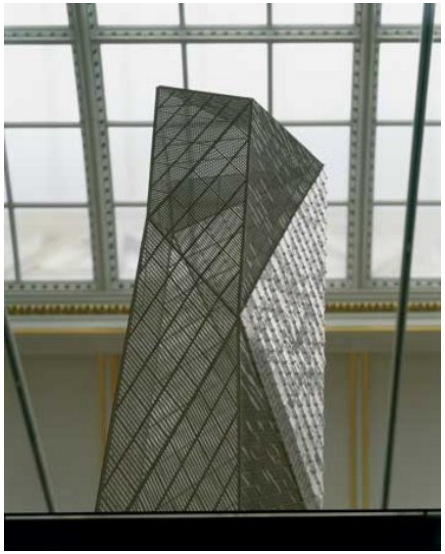
Eksekverbar Formell Modellering

- Modellering: lage en modell av systemdesign før implementasjon
- Formell: mulig å utlede egenskaper/konsekvenser av modellen
- Eksekverbar: la datamaskinen analysere modellen

Modellering

Modellering

Modellering

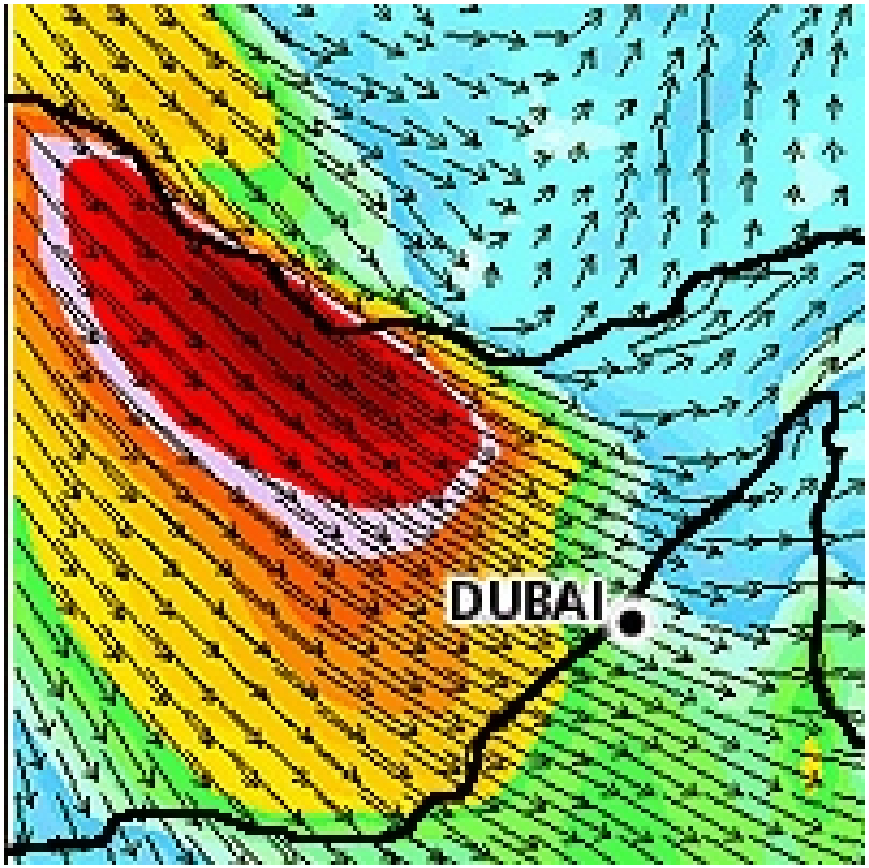


Modellering

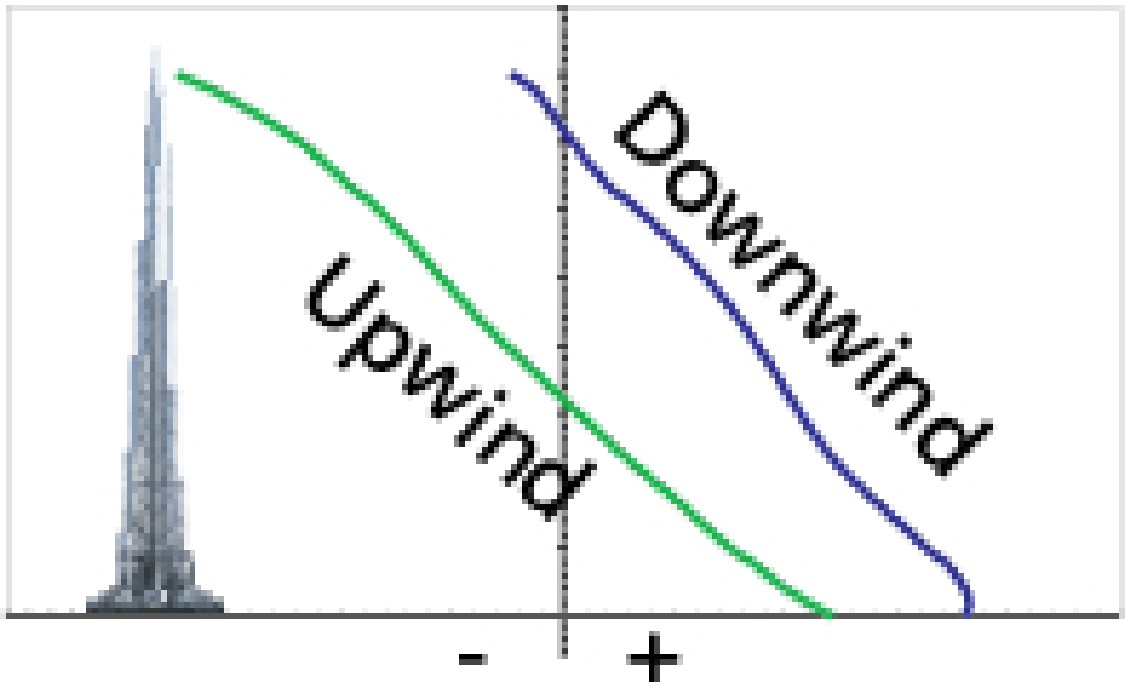
Modellering


(i vindtunnel)

Modellering



Modellering



Ifi tilbyr et videregående kurs i eksekverbar formell modellering:

INF 3230:

Formell modellering og analyse av kommuniserende systemer

- Modellene skrives i et state-of-the-art funksjonelt språk (Maude)
- Modellene kan analyseres direkte i Maude
- Modellene er funksjonelle programmer
- Ikke noe fikleri! Analysere design!

Ifi tilbyr et videregående kurs i eksekverbar formell modellering:

INF 3230:

Formell modellering og analyse av kommuniserende systemer

- Modellene skrives i et state-of-the-art funksjonelt språk (Maude)
- Modellene kan analyseres direkte i Maude
- Modellene er funksjonelle programmer
- Ikke noe fikleri! Analysere design!

Ifi tilbyr et videregående kurs i eksekverbar formell modellering:

INF 3230:

Formell modellering og analyse av kommuniserende systemer

- Modellene skrives i et state-of-the-art funksjonelt språk (Maude)
- Modellene kan analyseres direkte i Maude
- Modellene er funksjonelle programmer
- Ikke noe fikleri! Analysere design!

Eksempel: Merge-sort

```
fmod MERGE-SORT is protecting LIST-INT .
  op mergeSort : List -> List .
  op merge : List List -> List [comm] .

  vars L L' : List .
  vars NEL NEL' : NeList .
  vars I J : Int .

  eq mergeSort(nil) = nil .
  eq mergeSort(I) = I .

  ceq mergeSort(NEL NEL') =
        merge(mergeSort(NEL), mergeSort(NEL'))
    if length(NEL) == length(NEL')
       or  length(NEL) == length(NEL') +1 .

  eq merge(nil, L) = L .
  ceq merge(I L, J L') = I merge(L, J L')  if I <= J .
endfm
```

Eksempel: Merge-sort

```
fmod MERGE-SORT is protecting LIST-INT .
  op mergeSort : List -> List .
  op merge : List List -> List [comm] .

  vars L L' : List .
  vars NEL NEL' : NeList .
  vars I J : Int .

  eq mergeSort(nil) = nil .
  eq mergeSort(I) = I .

  ceq mergeSort(NEL NEL') =
        merge(mergeSort(NEL), mergeSort(NEL'))
    if length(NEL) == length(NEL')
       or  length(NEL) == length(NEL') +1 .

  eq merge(nil, L) = L .
  ceq merge(I L, J L') = I merge(L, J L')  if I <= J .
endfm
```

Eksempel: Merge-sort

```
fmod MERGE-SORT is protecting LIST-INT .
  op mergeSort : List -> List .
  op merge : List List -> List [comm] .

  vars L L' : List .
  vars NEL NEL' : NeList .
  vars I J : Int .

  eq mergeSort(nil) = nil .
  eq mergeSort(I) = I .

  ceq mergeSort(NEL NEL') =
          merge(mergeSort(NEL), mergeSort(NEL'))
      if length(NEL) == length(NEL')
         or  length(NEL) == length(NEL') +1 .

  eq merge(nil, L) = L .
  ceq merge(I L, J L') = I merge(L, J L')  if I <= J .
endfm
```

Eksempel: Merge-sort

```
fmod MERGE-SORT is protecting LIST-INT .
  op mergeSort : List -> List .
  op merge : List List -> List [comm] .

  vars L L' : List .
  vars NEL NEL' : NeList .
  vars I J : Int .

  eq mergeSort(nil) = nil .
  eq mergeSort(I) = I .

  ceq mergeSort(NEL NEL') =
          merge(mergeSort(NEL), mergeSort(NEL'))
      if length(NEL) == length(NEL')
         or  length(NEL) == length(NEL') +1 .

  eq merge(nil, L) = L .
  ceq merge(I L, J L') = I merge(L, J L')  if I <= J .
endfm
```

Videregående modellering: noen applikasjoner

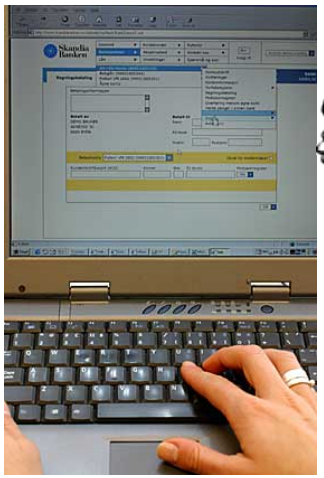
- Sliding windows protokollen
- Protokoller for distribuerte databaser
- Sikkerhetsprotokoller:
  - modellere klassiske sikkerhetsprotokoller
  - bruke Maude til å finne ut hvorvidt B-gjengen kan lure banken til overføre Skrue's penger

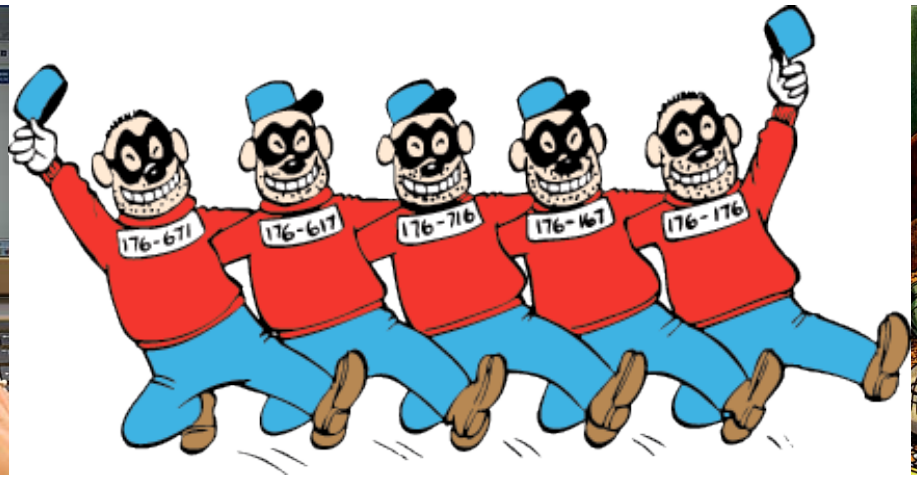
Videregående modellering: noen applikasjoner

- Sliding windows protokollen
- Protokoller for distribuerte databaser
- Sikkerhetsprotokoller:
  - modellere klassiske sikkerhetsprotokoller
  - bruke Maude til å finne ut hvorvidt B-gjengen kan lure banken til overføre Skrue's penger

Videregående modellering: noen applikasjoner

- Sliding windows protokollen
- Protokoller for distribuerte databaser
- Sikkerhetsprotokoller:
  - modellere klassiske sikkerhetsprotokoller
  - bruke Maude til å finne ut hvorvidt B-gjengen kan lure banken til overføre Skrue's penger





Forhåndskunnskaper

- Ingen spesielle forhåndskrav
- Fordel å ha programmert noe rekursivt
- Kurset er også teoretisk
  - godt å vite hva en funksjon $f : A \rightarrow B$ er
  - lærer å analysere hvorvidt programmer har evige løkker

Velkommen til INF 3230 våren 2013!

Forhåndskunnskaper

- Ingen spesielle forhåndskrav
- Fordel å ha programmert noe rekursivt
- Kurset er også teoretisk
  - godt å vite hva en funksjon $f : A \rightarrow B$ er
  - lærer å analysere hvorvidt programmer har evige løkker

Velkommen til INF 3230 våren 2013!