

1 Forelesning 17

1.1 Pakke og byggesystemer

Problem:

- Kompilatoren kompilerer programmet
- Hvert program kan avhenge av flere biblioteker
- For at programmet skal kompilere må alle bibliotekene som det avhenger av lastes ned og kompileres først.

Å gjøre denne nedlastingen og kompileringen manuelt er mye arbeid. Et pakke- og byggesystem forsøker å løse dette.

1.1.1 Hva er et programbyggesystem?

Vanlig ontologi for et pakke- og bygge system

Pakke: En helhetlig samling med kildekodefiler som kompileres til et eller flere programmer/bibliotek

Bygge en pakke: Kompilere kildekoden og sette sammen programmene/bibliotekent.

1.1.2 Pakkeversjoner

Programvare er i stadig utvikling. Derfor kommer pakker i ulike versjoner

- Noen ganger er pakke B bare kompatibel med noen versjoner av pakke A.

Et pakkesystem må sørge for at det ikke blir konflikter når ulike pakker krever ulike versjoner av en avhengighet.

1.1.3 Eksempler på pakke- og byggesystemer

Språktilknyttede:

- Maven (java)
- Pip (Python)
- Cabal (Haskell)
- npm (JavaScript)

Operativsystemtilknyttede:

- Make (*nix, språkdiagnostisk)
- Ports, portage, nix
- Homebrew (MacOS)

1.1.4 Cabal

Cabal er et pakke- og byggesystem for Haskellprogrammer

- En pakke beskrives av en `.cabal` fil.
- Hackage er et sentralt nettsted for offentlige pakker

1.1.5 Cabal Hell

Cabal har alltid hatt god støtte for å spesifisere hvilke pakker som er kompatible med programmet.

- Tidligere var det kun mulig å ha en versjon av en pakke installert.
- Det var veldig lett å havne i en situasjon hvor pakker var i konflikt med hverandre.

1.1.6 Stack

Stack er et alternativt system til cabal-install

- Bruker cabal-biblioteket.
- Har et annet sentralt pakkeoppsamlingssted
- Har en annen strategi for å løse "cabal hell"
- Tar også ansvar for GHC installasjonen

Stack har et begrep om *snapshot*; en konsistent samling pakker som ikke er i konflikt, sammen med en spesifikk GHC versjon.

1.2 Testing

Det finnes mange former for testing:

- Unit testing
- End-toend testing
- Statisk / dynamisk testing
- Integrasjonstesting, systemtesting

QuickCheck er et verktøy for property testing.

1.2.1 Egenskapstesting

I property testing så tester man at funksjoner har gitte egenskaper

- Hver egenskap er en funksjon
- Funksjonen tar argumenter og tester om en gitt egenskap holder (Boolsk verdi)
- Testene kjøres på tilfeldig generert data