QuviQ

# Property-based Testing

Hans Svensson

SAST Väst – Q4

2013-11-26

http://quviq.com/

# Om mig

- Teknisk Fysik
- PhD Datavetenskap
  - Verification of Erlang programs using: Testing, Model checking, and Theorem proving
- PostDoc
  - Property-based testing (PROTEST)
- QuviQ
  - Forskning
  - Utveckling av QuviQ QuickCheck
  - Modellering
  - Utbildning

# Exempel – att vända på en lista

- Vi vill testa funktionen **`reverse`**:
  - Funktionen tar ett argument, en lista
  - Funktionen returnerar den omvända listan

Ex:

```
reverse([7, 3, 5, 8]) -> [8, 5, 3, 7]
```

# Exempel – att vända på en lista

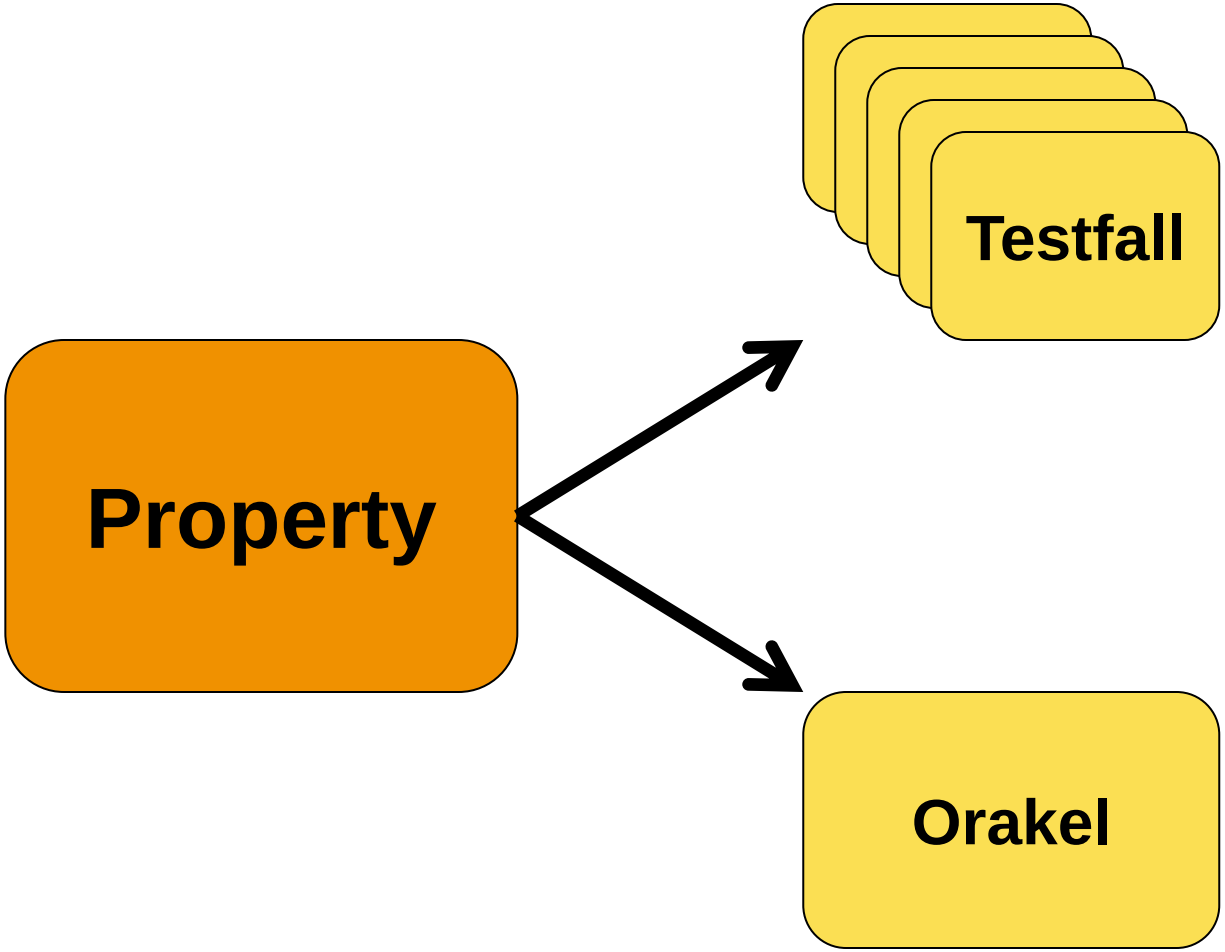
- Det vanligaste sättet – några enhetstester:

```
empty_test() -> ?ASSERT([], reverse([])).

single_test() -> ?ASSERT([1], reverse([1]).

list1_test() ->
  ?ASSERT([1, 4, 9], reverse([9, 4, 1]).

list1_test() ->
  ?ASSERT([7, 3, 5, 8], reverse([8, 5, 3, 7]).
...
```

# Exempel – att vända på en lista

- Istället för testfall – tänk på egenskaper
  - Den returnerade listan skall ha lika många element
  - Varje element i den ursprungliga listan måste finnas i den resulterande listan
  - Om man reverserar en lista två gånger är resultatet identiskt med den ursprungliga listan

# Property-based testing



# Exempel – att vända på en lista

```
empty_test() -> ?ASSERT([], reverse([])).

single_test() -> ?ASSERT([1], reverse([1]).

list1_test() ->
  ?ASSERT([1, 4, 9], reverse([9, 4, 1]).

list1_test() ->
  ?ASSERT([7, 3, 5, 8], reverse([8, 5, 3, 7]).
...
```

# Exempel – att vända på en lista

```
tests() ->
  [[], [1], [9, 4, 1], [8, 5, 3, 7]].

foreach(
  fun(T) ->
    ?ASSERT(T, reverse(reverse(T)))
  end, tests()).
```

Skulle det inte vara bättre att testa fler listor (längre?)

För varje lista, kontrollera att listan är identisk med resultatet av reverse(reverse(Lista)).

# Exempel – att vända på en lista

```
tests(0) -> [];
tests(N) -> [ random_list() | tests(N - 1)].

random_list() ->
  random_list(random(1, ?MAX_LEN)).
random_list(0) -> [];
random_list(N) ->
  [random(-?MAXN, ?MAXN) | random_list(N-1)].

foreach(fun(T) ->
    ?ASSERT(T, reverse(reverse(T)))
  end, tests(10)).
```

A bit ad hoc!?

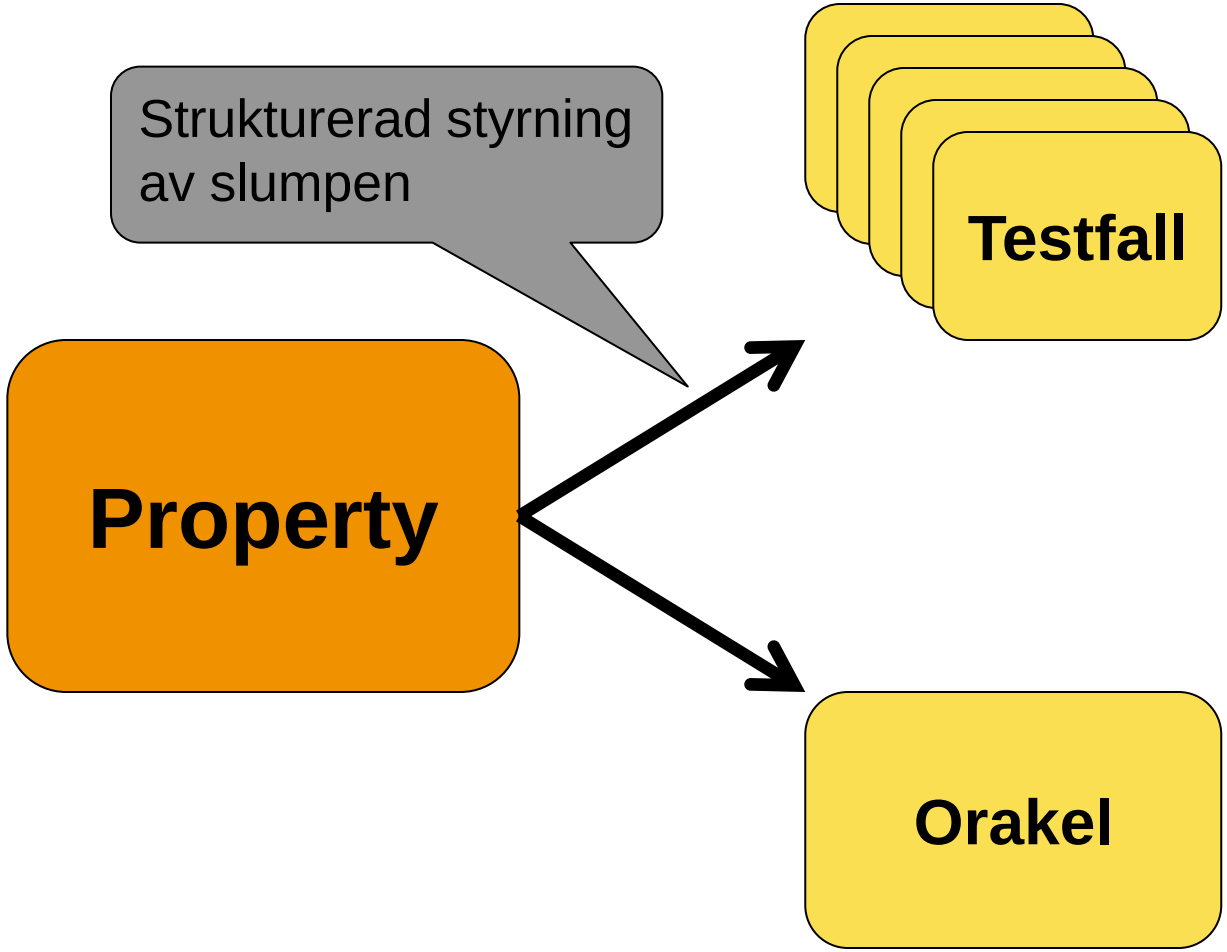
Nu är det enkelt att testa fler listor...

# Exempel – att vända på en lista

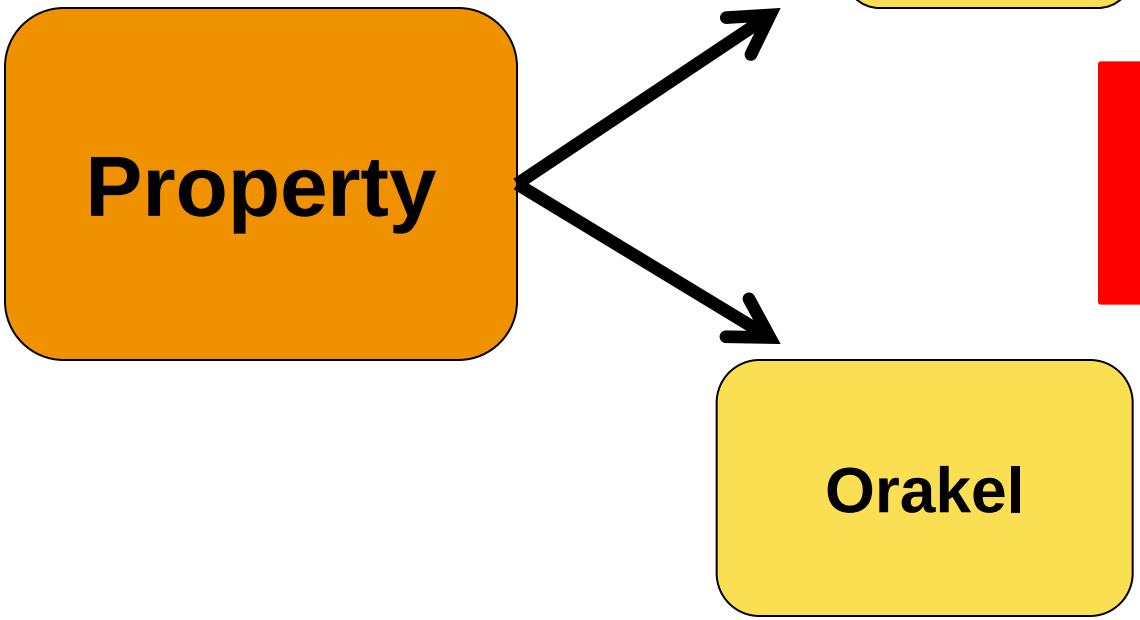
- Kan vi uttrycka samma sak på ett mer strukturerat och enklare sätt?

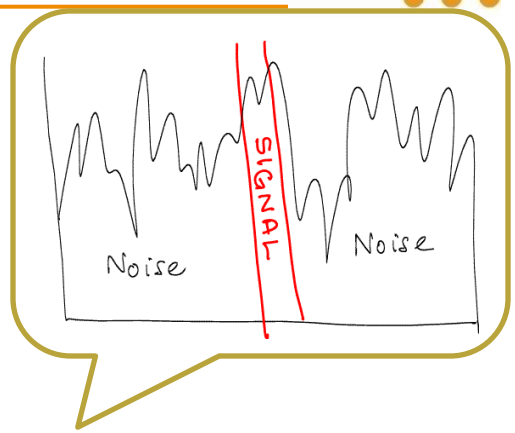
```
prop_reverse() ->
  ?FORALL(List, list(int()),
     List == reverse(reverse(List)).
```

# PBT 2.0

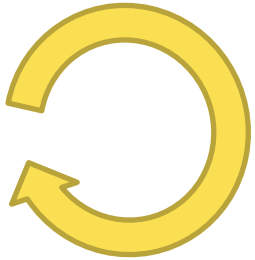

Strukturerad styrning
av slumpen
Testfall
Property
Orakel

# PBT 2.0




Minimalt testfall



Shrinking

# PBT - Fördelar

- En modell många testfall
- Lättare (billigare!) att underhålla – mindre kod
- En modell flera olika typer av testfall
- Shrinking ger enklare analys

# PBT - Nackdelar

Inte nödvändigtvis en nackdel!?

- Svårare att skriva/utveckla
- Explicita testfall enklare att förstå (och förklara för management!)
- Vad har vi testat?

Enklare att svara på i traditionell testning, men förmodligen har vi testat mindre!

# PBT i verkligheten

- Skalar det? Ja.
- Var har vi använt PBT:
  - Telekomindustrin
  - Automotivetillämpningar – AUTOSAR
  - Distribuerade Key-Value system
  - Finansapplikationer
  - Websystem

# Överlappande koncept

- Vad skiljer PBT från MBT (Model-based testing)
  - Det beror på vem du frågar!
  - Traditionell MBT strävar efter att hitta en minimal testsvit som uppfyller vissa kriterier

# Mentalitet

- Vi har sett väldigt skiftande mentalitet när det gäller upptäckta buggar!
  - Du hittade en bugg, bra!
  - Nej, hittade du en bugg!?
  - Vi har sett mycket liten korrelation till bransch/domän, kostnad för fel, teamstorlek, etc.

# Forskning i industrin

- QuviQ är ett avknoppningsföretag från Chalmers
- För närvarande gör jag ungefär 50% forskning
- Korsbefruktning – utveckling och forskning
- Snabbare återkoppling i industrin
- Quviqs forskningsarbete
  - Partner i tre större forskningsprojekt
  - Handledning av examensarbeten
  - Programkommitteer, workshops, etc

# Forskningsprojekt

- Prowess
  - Property-based testing for web services

- Acsäpt
  - Acceptanstest av säkerhetskritisk plattformsprogramvara

- nSafeCer
  - processer och metoder för att återanvända säkerhetsargumentation

# Prowess


- EU - FP 7 projekt
- 9 partners - Sverige, UK, Spanien
- PBT applicerat på Web services
  - Kompositionella modeller
  - Icke-funktionella krav
  - Kvalitet och kvalitetsmått för PBT
  - Pilotstudier i industrin

http://prowess-project.eu/

# Acsäpt

- Del av Vinnovas FFI-program
- Svenskt projekt – SP, QuviQ, Volvo och Mecel .
- Acceptanstestkriterer för plattformsprogramvara
  - Fokus på AUTOSAR
  - ISO 26262 i fordonsindustrin
  - Svar på frågan: ***Vad har vi testat?***

# Fler exempel – AUTOSAR

- CAN – kommunikationsbuss i bilen
  - 4 x ~150 sidor specifikation (+ ISO standarder)
  - ~4000 rader kod i modellen

  - Prioriterad kommunikation (lågt Id – hög prioritet)
  - Den testade implementationen prioriterade tvärtom!
  - Deras enhetstester prioriterade tvärtom!!

# Fler exempel – Telekom

- Vi testade en samtalshanterande komponent
- Testfallen som genererades var sekvenser av
  - add, remove, och terminate
- Max två samtalande parter i ett samtal
- Ett testfall som innehöll 160 kommandon gav ett fel
- Efter shrinkning: 7 kommandon
- Add – Add – Remove – Add – Remove – Add – Remove - <Krasch>

Vem skulle skriva detta testfall manuellt?