GETTING SOFTWARE RIGHT WITH PROPERTIES, GENERATED TESTS, AND PROOFS

Evolve your hack into robust software

Michael Sperber

Active Group GmbH

@sperbsen

BERLIN, FEBRUARY 28

https://bobkonf.de/

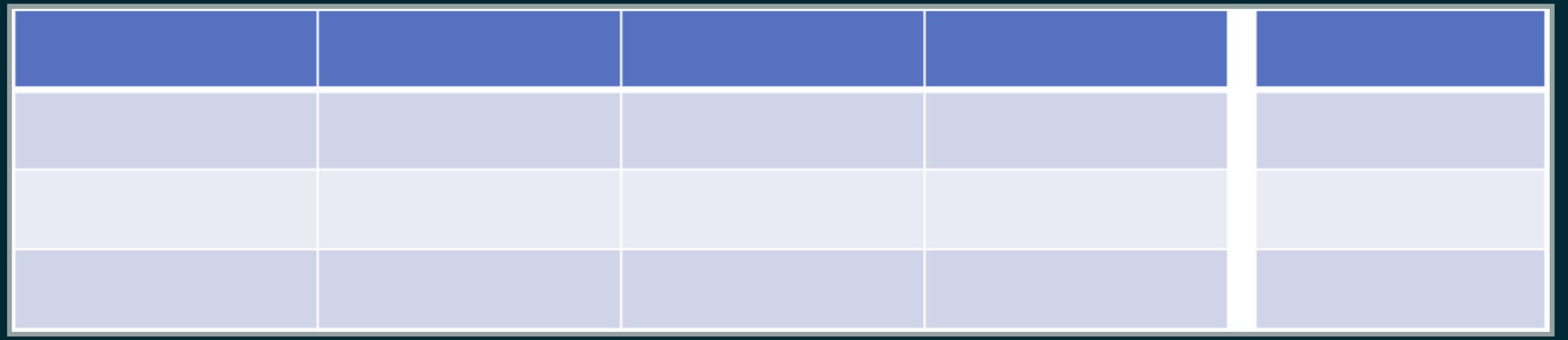
INTRODUCTORY TALK!

ANIMALS ON THE TEXAS HIGHWAY

```
data Liveness = Dead | Alive

Weight : Type
Weight = Int

data Animal : Type where
  Dillo : Liveness -> Weight -> Animal
  Parrot : String -> Weight -> Animal

a1 : Animal
a1 = Dillo Alive 10
a2 : Animal
a2 = Dillo Dead 12
a3 : Animal
a3 = Parrot "The treasure is on treasure island!" 3

runOverAnimal : Animal -> Animal
runOverAnimal (Dillo liveness weight) = Dillo Dead weight
runOverAnimal (Parrot sentence weight) = Parrot "" weight
```

DOMAIN MODELS GROW

DOMAIN MODELS GROW



WHAT'S THIS?

$$(a + b) + c = a + (b + c)$$

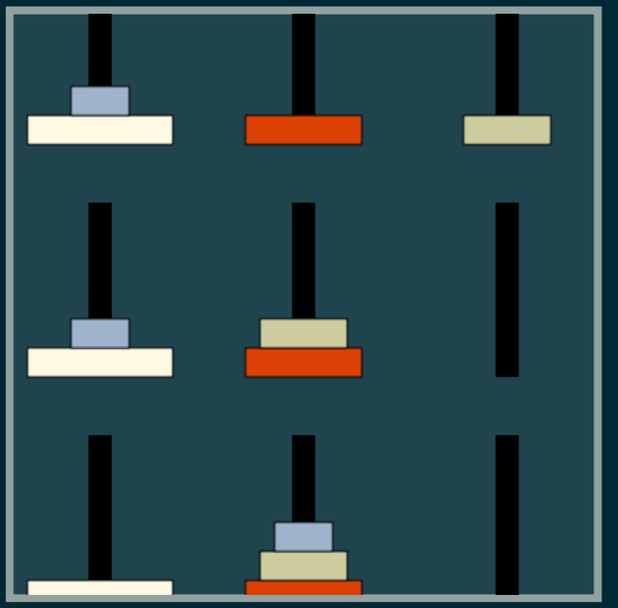
NUMBERS AND ADDITION

$$\forall a, b, c \in \mathbb{N} : (a + b) + c = a + (b + c)$$

LISTS AND CONCATENATION

$$\forall a, b, c \in \texttt{List}\ el : a \mathbin{+\!\!+} (b \mathbin{+\!\!+} c) = (a \mathbin{+\!\!+} b) \mathbin{+\!\!+} c$$

IMAGES



(source: Brent Yorgey)

IMAGES

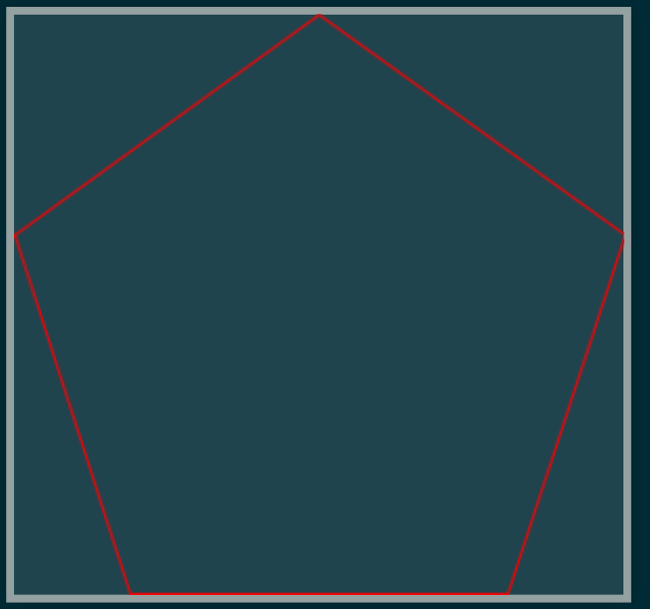
```
data Image = ...
```

STAR

```
star : Int -> Mode -> Color -> Image

goldStar : Image
goldStar = star 200 Solid Gold
```

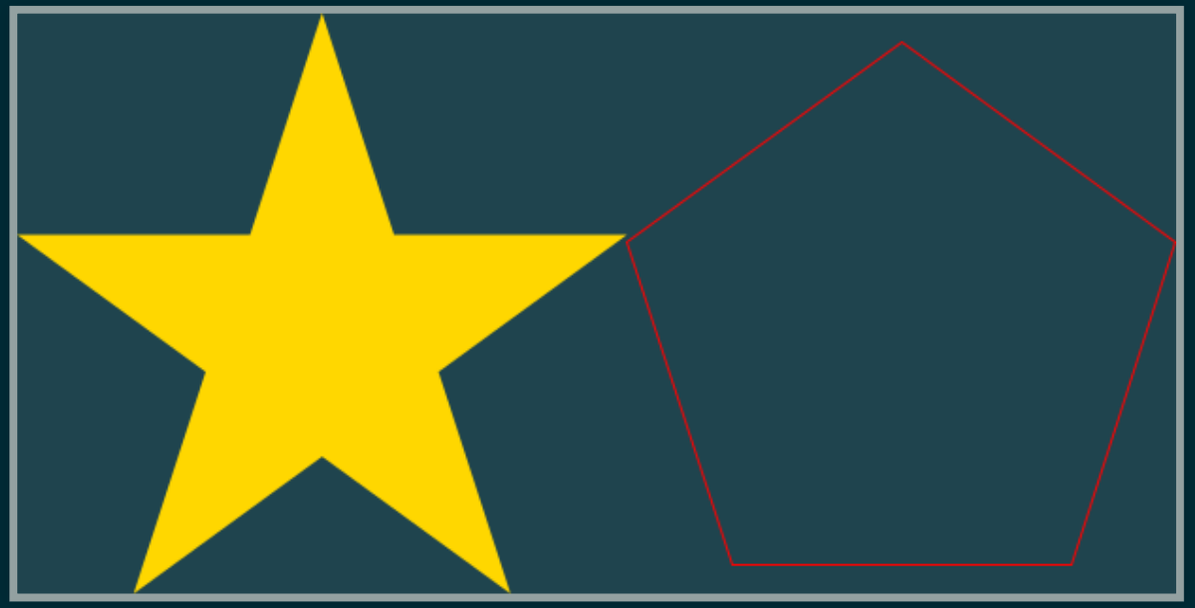
POLYGON

```
polygon : Int -> Int -> Mode -> Color -> Image

pentagon : Image
pentagon = polygon 180 5 Outline Red
```



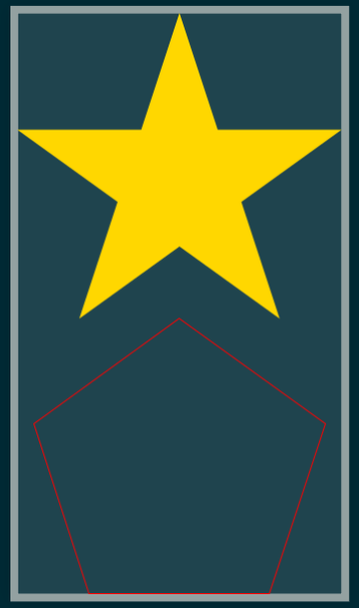
BESIDE

```
beside : Image -> Image -> Image

beside goldStar pentagon
```



ABOVE

```
above : Image -> Image -> Image

above goldStar pentagon
```



COMBINATION

```
above (beside goldStar pentagon)
      (beside pentagon goldStar)
```

OVERLAY

```
overlay : Image -> Image -> Image

overlay goldStar pentagon
```

ASSOCIATIVITY FOR IMAGES

$$\forall a, b, c \in \texttt{Image} :$$
$$\texttt{overlay}\ (\texttt{overlay}\ a\ b)\ c) = \texttt{overlay}\ a\ (\texttt{overlay}\ b\ c)$$

SEMIGROUP

set S

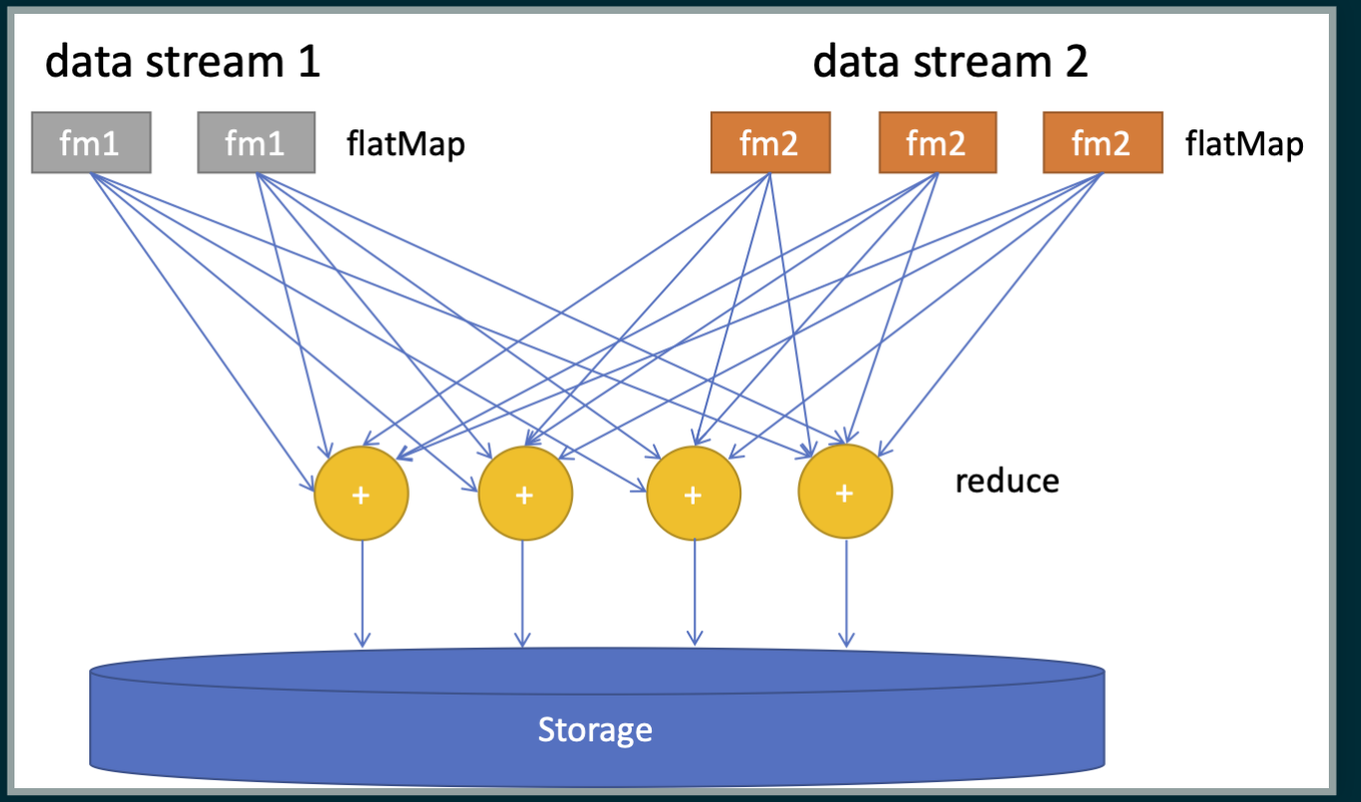
$$\circ : S \to S \to S$$

$$\forall a, b, c \in S : (a \circ b) \circ c = a \circ (b \circ c)$$

PARENTHESES DON'T MATTER

$$(a \circ (b \circ (c \circ d))) \circ (e \circ f) = a \circ b \circ c \circ d \circ e \circ f$$

DISTRIBUTED COMPUTATION



DESIGN

Monoids: Theme and Variations *(Functional Pearl)*

Brent A. Yorgey
University of Pennsylvania
byorgey@cis.upenn.edu

Abstract

The *monoid* is a humble algebraic structure, at first glance even downright boring. However, there's much more to monoids than meets the eye. Using examples taken from the diagrams vector graphics framework as a case study, I demonstrate the power and beauty of monoids for library design. The paper begins with an extremely simple model of diagrams and proceeds through a series of incremental variations, all related somehow to the central theme of monoids. Along the way, I illustrate the power of compositional semantics; why you should also pay attention to the monoid's even humbler cousin, the *semigroup*; monoid homomorphisms; and monoid actions.

Categories and Subject Descriptors D.1.1 [*Programming Techniques*]: Applicative (Functional) Programming; D.2.2 [*Design Tools and Techniques*]

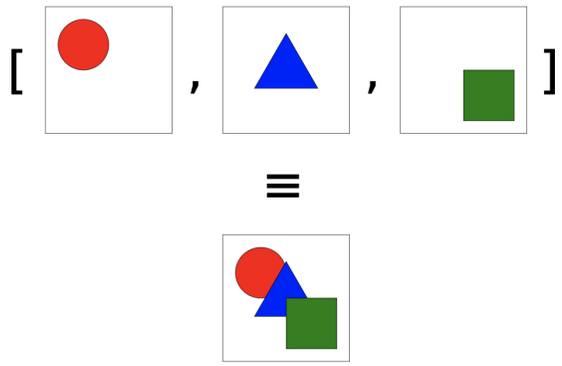


Figure 1. Superimposing a list of primitives

B. Yorgey: Monoids: Theme and Variations

MONOID

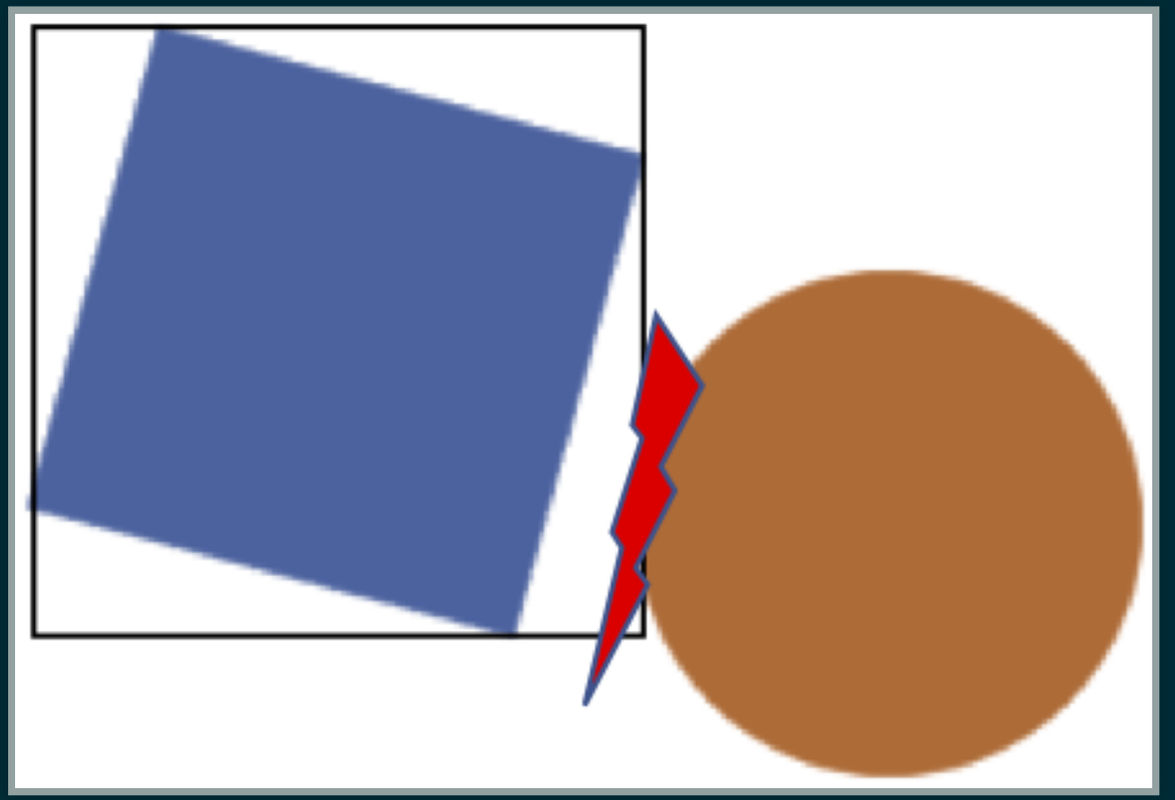
Semigroup and …

$$\begin{aligned} \mathbf{n} &\in S \\ \forall a \in S : a \circ \mathbf{n} &= \mathbf{n} \circ a = a \end{aligned}$$

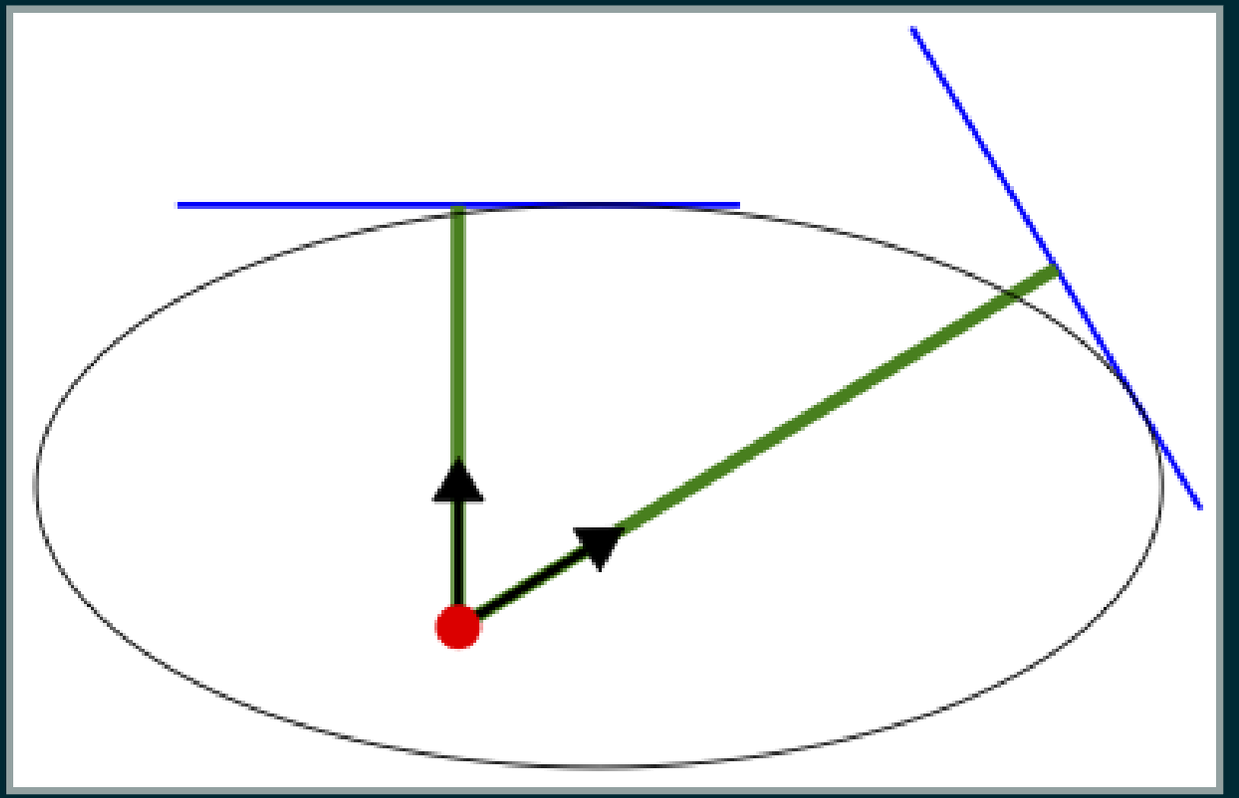
MONOIDS IN THE WILD

- numbers
- lists
- images
- music
- animations
- financial contracts
- semiconductor-fabrication routes
- properties
- pretty printers
- …

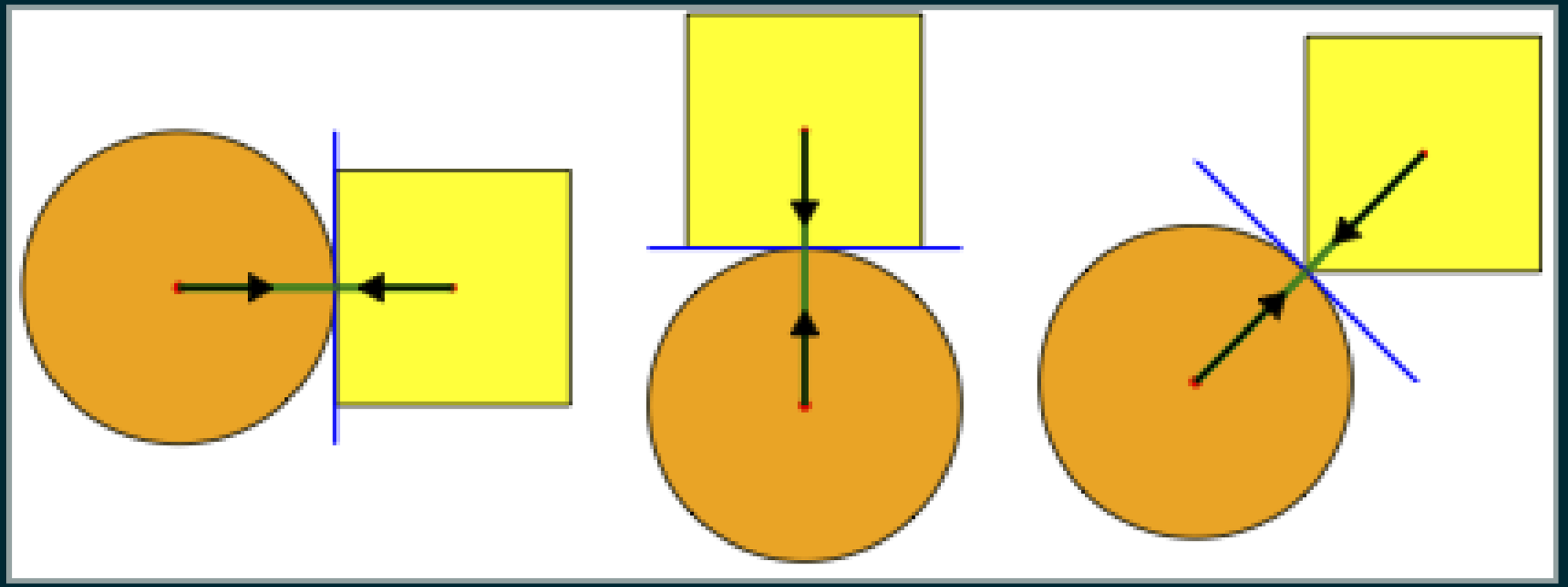
BOUNDING BOX PROBLEM



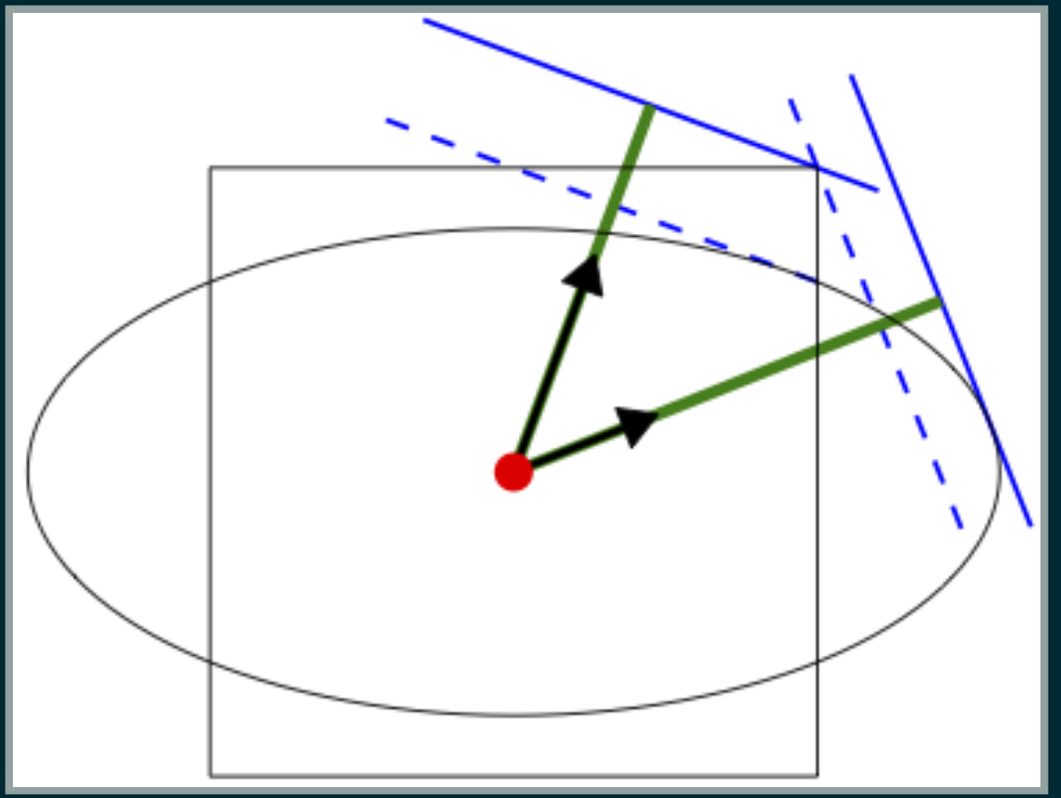
ENVELOPES



COMPOSING WITH ENVELOPES



COMPOSING ENVELOPES



ASSOCIATIVITY

$$\forall \texttt{image1}, \texttt{image2}, \texttt{image3} \in \texttt{Image}.$$

```
overlay (overlay image1 image2) image3 == overlay image1 (overlay image2 image3)
```

ASSOCIATIVITY

```
prop_overlayAssociative =
  forAll (arbTriple arbImage arbImage arbImage) (\ image1 image2 image3 =>
    overlay (overlay image1 image2) image3 == overlay image1 (overlay image2 image3))
```

QUICKCHECK

John Hughes

https://www.chalmers.se/

INTERVAL SETS

```
ISet : Type
ISet = List (Nat, Nat)

iToList : ISet -> List Nat

λΠ> iToList [(0, 3), (5, 7), (9, 10)]
[0, 1, 2, 3, 5, 6, 7, 9, 10] : List Nat
```

VALIDITY

```
isValid : ISet -> Bool

isValid [] = True

isValid [(lo, hi)] = lo <= hi

isValid ((lo1, hi1) :: (lo2, hi2) :: rest) =
  (lo1 <= hi1) && (hi1+1 < lo2) && isValid ((lo2, hi2)::rest)
```

UNION

```
iUnion : ISet -> ISet -> ISet
```

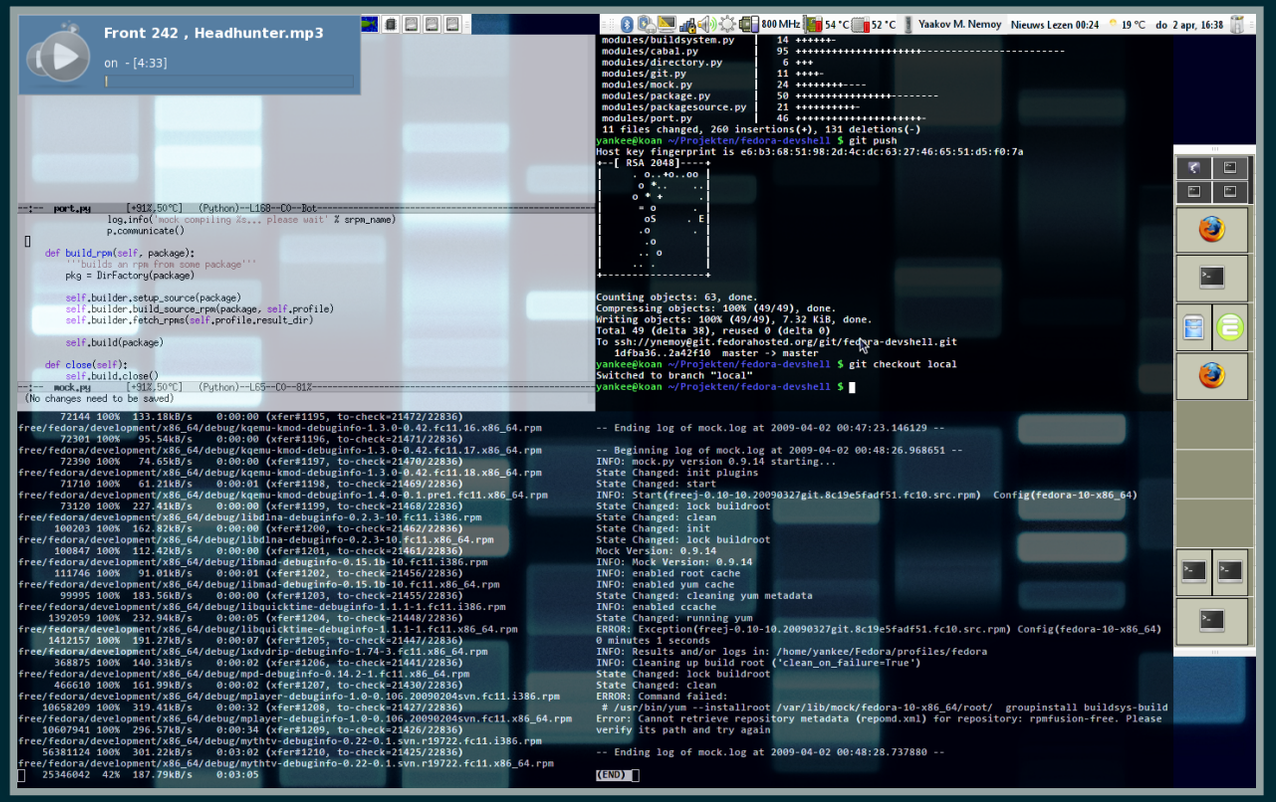
SIMPLE CRITERION

```
prop_unionValid =
  forAll (arbPair arbISet arbISet) (\ (iset1, iset2) =>
    isValid (iUnion iset1 iset2))
```

TEST

```
prop_unionCorrect =
  forAll (arbPair arbISet arbISet) (\ (iset1, iset2) =>
    iToList (iUnion iset1 iset2)
    ==
    merge2 (iToList iset1) (iToList iset2))
```

XMONAD


Front 242 , Headhunter.mp3
on - [4:33]
Yaakov M. Nemoy
Nieuws Lezen 00:24
do 2 apr, 16:38
modules/buildsystem.py | 14 ++++++-
modules/cabal.py | 95 ++++++++++++++++++++++++++++----------------------
modules/directory.py | 6 +++
modules/git.py | 11 ++++++-
modules/mock.py | 24 ++++++++----
modules/package.py | 50 +++++++++++++++++++++++++++
modules/packagesource.py | 21 ++++++++++
modules/port.py | 46 +++++++++++++++++++++++++
11 files changed, 260 insertions(+), 131 deletions(-)
yankee@koan ~/Projekten/fedora-devshell $ git push
Host key fingerprint is e6:b3:68:51:98:2d:4c:dc:63:27:46:65:51:d5:f0:7a
+--[RSA 2048]----+
Counting objects: 63, done.
Compressing objects: 100% (49/49), done.
Writing objects: 100% (49/49), 7.32 KiB, done.
Total 49 (delta 38), reused 0 (delta 0)
To ssh://ynemoy@git.fedorahosted.org/git/fedora-devshell.git
1dfba36..2a42f10 master -> master
yankee@koan ~/Projekten/fedora-devshell $ git checkout local
Switched to branch "local"
yankee@koan ~/Projekten/fedora-devshell $
port.py [+91%,50°C] (Python)--L168--C0--Bot
log.info('mock compiling %s... please wait' % srpm_name)
p.communicate()
def build_rpm(self, package):
'''builds an rpm from some package'''
pkg = DirFactory(package)
self.builder.setup_source(package)
self.builder.build_source_rpm(package, self.profile)
self.builder.fetch_rpms(self.profile.result_dir)
self.build(package)
def close(self):
self.build.close()
mock.py [+91%,50°C] (Python)--L65--C0--81%
(No changes need to be saved)
72144 100% 133.18kB/s 0:00:00 (xfer#1195, to-check=21472/22836)
free/fedora/development/x86_64/debug/kqemu-kmod-debuginfo-1.3.0-0.42.fc11.16.x86_64.rpm
72301 100% 95.54kB/s 0:00:00 (xfer#1196, to-check=21471/22836)
free/fedora/development/x86_64/debug/kqemu-kmod-debuginfo-1.3.0-0.42.fc11.17.x86_64.rpm
72390 100% 74.65kB/s 0:00:00 (xfer#1197, to-check=21470/22836)
free/fedora/development/x86_64/debug/kqemu-kmod-debuginfo-1.3.0-0.42.fc11.18.x86_64.rpm
71710 100% 61.21kB/s 0:00:01 (xfer#1198, to-check=21469/22836)
free/fedora/development/x86_64/debug/kqemu-kmod-debuginfo-1.4.0-0.1.pre1.fc11.x86_64.rpm
73120 100% 227.41kB/s 0:00:00 (xfer#1199, to-check=21468/22836)
free/fedora/development/x86_64/debug/libdlna-debuginfo-0.2.3-10.fc11.i386.rpm
100203 100% 162.82kB/s 0:00:00 (xfer#1200, to-check=21462/22836)
free/fedora/development/x86_64/debug/libdlna-debuginfo-0.2.3-10.fc11.x86_64.rpm
100847 100% 112.42kB/s 0:00:00 (xfer#1201, to-check=21461/22836)
free/fedora/development/x86_64/debug/libmad-debuginfo-0.15.1b-10.fc11.i386.rpm
111746 100% 91.01kB/s 0:00:01 (xfer#1202, to-check=21456/22836)
free/fedora/development/x86_64/debug/libmad-debuginfo-0.15.1b-10.fc11.x86_64.rpm
99995 100% 183.56kB/s 0:00:00 (xfer#1203, to-check=21455/22836)
free/fedora/development/x86_64/debug/libquicktime-debuginfo-1.1.1-1.fc11.i386.rpm
1392059 100% 232.94kB/s 0:00:05 (xfer#1204, to-check=21448/22836)
free/fedora/development/x86_64/debug/libquicktime-debuginfo-1.1.1-1.fc11.x86_64.rpm
1412157 100% 191.27kB/s 0:00:07 (xfer#1205, to-check=21447/22836)
free/fedora/development/x86_64/debug/lxdvdrip-debuginfo-1.74-3.fc11.x86_64.rpm
368875 100% 140.33kB/s 0:00:02 (xfer#1206, to-check=21441/22836)
free/fedora/development/x86_64/debug/mpd-debuginfo-0.14.2-1.fc11.x86_64.rpm
466610 100% 161.99kB/s 0:00:02 (xfer#1207, to-check=21430/22836)
free/fedora/development/x86_64/debug/mplayer-debuginfo-1.0-0.106.20090204svn.fc11.i386.rpm
10658209 100% 319.41kB/s 0:00:32 (xfer#1208, to-check=21427/22836)
free/fedora/development/x86_64/debug/mplayer-debuginfo-1.0-0.106.20090204svn.fc11.x86_64.rpm
10607941 100% 296.57kB/s 0:00:34 (xfer#1209, to-check=21426/22836)
free/fedora/development/x86_64/debug/mythtv-debuginfo-0.22-0.1.svn.r19722.fc11.i386.rpm
56381124 100% 301.22kB/s 0:03:02 (xfer#1210, to-check=21425/22836)
free/fedora/development/x86_64/debug/mythtv-debuginfo-0.22-0.1.svn.r19722.fc11.x86_64.rpm
25346042 42% 187.79kB/s 0:03:05
-- Ending log of mock.log at 2009-04-02 00:47:23.146129 --
-- Beginning log of mock.log at 2009-04-02 00:48:26.968651 --
INFO: mock.py version 0.9.14 starting...
State Changed: init plugins
State Changed: start
INFO: Start(freej-0.10-10.20090327git.8c19e5fadf51.fc10.src.rpm) Config(fedora-10-x86_64)
State Changed: lock buildroot
State Changed: clean
State Changed: init
State Changed: lock buildroot
Mock Version: 0.9.14
INFO: Mock Version: 0.9.14
INFO: enabled root cache
INFO: enabled yum cache
State Changed: cleaning yum metadata
INFO: enabled ccache
State Changed: running yum
ERROR: Exception(freej-0.10-10.20090327git.8c19e5fadf51.fc10.src.rpm) Config(fedora-10-x86_64)
0 minutes 1 seconds
INFO: Results and/or logs in: /home/yankee/Fedora/profiles/fedora
INFO: Cleaning up build root ('clean_on_failure=True')
State Changed: lock buildroot
State Changed: clean
ERROR: Command failed:
/usr/bin/yum --installroot /var/lib/mock/fedora-10-x86_64/root/ groupinstall buildsys-build
Error: Cannot retrieve repository metadata (repomd.xml) for repository: rpmfusion-free. Please verify its path and try again
-- Ending log of mock.log at 2009-04-02 00:48:28.737880 --
(END)

XMONAD

```
record StackSet (window : Type)
  constructor StackSet
  current : Int
  stacks : Map Int (List window)
```

(source: Don Stewart)

OPERATIONS

```
empty  : Nat -> StackSet window
view  : Nat -> StackSet window -> StackSet window
peek  : StackSet window -> Maybe window
rotate  : Ordering -> StackSet window -> StackSet window
push  : window -> StackSet window -> StackSet window
insert  : window -> Nat -> StackSet window -> StackSet window
delete  : window -> StackSet window -> StackSet window
shift  : Nat -> StackSet window -> StackSet window
```

INVARIANT

```
invariant : StackSet window -> Bool
invariant stackSet =
    let windows = windowList stackSet
    in (current stackSet < Map.size (stacks stackSet)) &&
       (removeDuplicates windows == windows)
```

INVARIANT

```
prop_empty_I = forAll (arbPair arbNat) (\ stackIndex  =>
  invariant (empty stackIndex))

prop_view_I = forAll (arbPair arbNat arbStackSet) (\ (stackIndex, stackSet) =>
  invariant (view stackIndex stackSet))

prop_rotate_I = forAll (arbPair arbNat arbStackSet) (\ (stackIndex, stackSet) =>
  invariant (rotate stackIndex stackSet))

prop_push_I = forAll (arbPair arbNat arbStackSet) (\ (stackIndex, stackSet) =>
  invariant (push stackIndex stackSet)

prop_delete_I = forAll (arbPair arbNat arbStackSet) (\ (stackIndex, stackSet) =>
  invariant (delete stackIndex stackSet)

prop_shift_I = forAll (arbPair arbNat arbStackSet) (\ (stackIndex, stackSet) =>
  stackIndex < size stackSet
    ==> invariant (shift stackIndex stackSet)

prop_insert_I = forAll (arbPair arbNat arbStackSet) (\ (stackIndex, stackSet) =>
  window < size stackSet
    ==> invariant (insert stackIndex window stackSet)
```

MIGRATING FROM VISUAL BASIC

```
Public Shared Function
  CheckHash(Password As String, Hash As String) As Boolean

Public Shared Function
  HashPassword(Password As String) As String
```

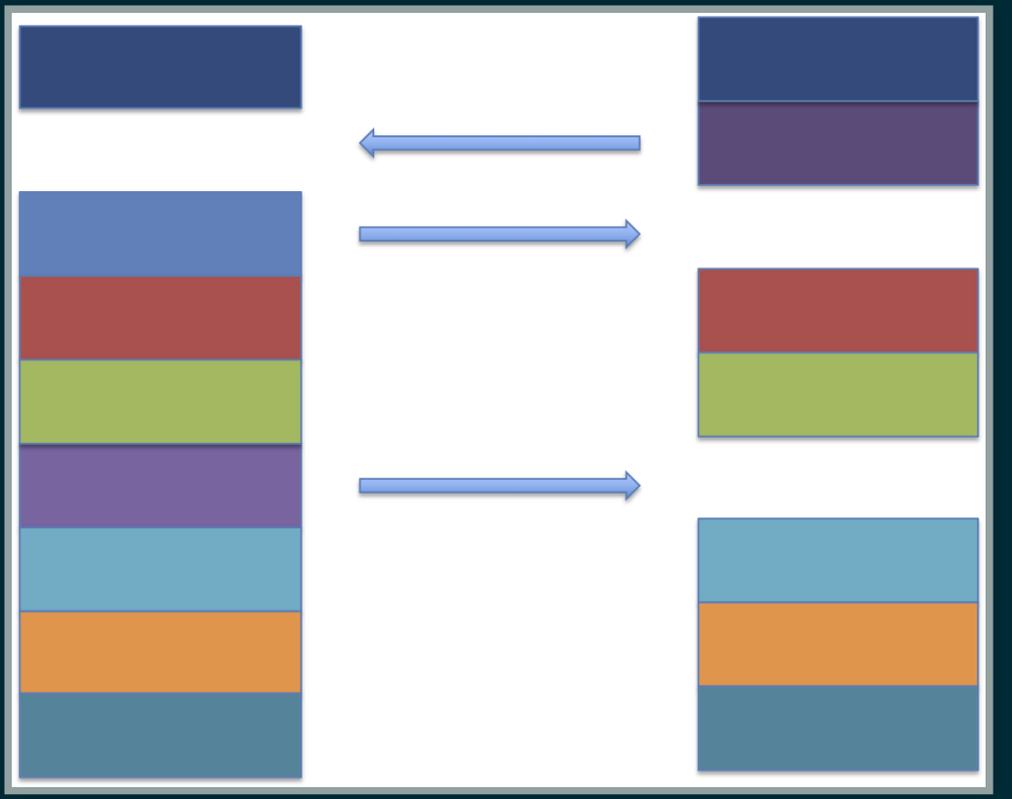
PROPERTY

```
prop_passwordCorrect =
  forAll arbString (\ password =>
     compareWithHash password
                    (createUnsaltedPseudoHash password))
```

SURPRISE

```
prop_passwordCorrectReally =
  forAll arbString (\ password =>
     compareWithHash password
                    (restrictTo11Chars
                      (createUnsaltedPseudoHash password)))
```

SYNCHRONIZATION



SYNCHRONIZATION PROPERTY

```
forAll (arbPair (arbSet arbBlock) (arbSet arbBlock)) (\ (bs1, bs2) =>
  let all = Set.union bs1 bs2
      (bs1', bs2') = synchronize (Set.toList bs1)
                                 (Set.toList bs2)
  in (Set.union bs1 bs1' == all) &&
     (Set.union bs2 bs2' == all) &&
     (Set.isEmpty (Set.intersect bs1 bs1')) &&
     (Set.isEmpty (Set.intersect bs2 bs2'))
```

MNESIA

```
Prefix:
open_file(dets_table ,[{type,bag}]) --> dets_table
close(dets_table) --> ok
open_file(dets_table ,[{type,bag}]) --> dets_table
Parallel:
1. lookup(dets_table ,0) --> []
2. insert(dets_table ,{0,0})
3. insert(dets_table ,{0,0})
Result: ok
```

J. Hughes: Experiences with QuickCheck: Testing the Hard Stuff and Staying Sane

DROPBOX

Client 1	Client 2
WRITE_1 ‘a’ $\rightarrow \perp$	
WRITE_1 $\perp \rightarrow$ ‘a’	
	WRITE_2 ‘b’ $\rightarrow$ ‘a’
WRITE_1 ‘c’ $\rightarrow \perp$	
$\text{READ}_1 \rightarrow \perp$	

J. Hughes et al.: Mysteries of Dropbox

SCREENCAST EDITOR

1. Timeline flattening
2. Video scene classification
3. Focus and timeline consistency
4. Symmetry of undo/redo

O. Wikstrom: Property-Based Testing in a Screencast Editor

PROOFS

```
(++) : List a -> List a -> List a
(++) []      right = right
(++) (x::xs) right = x :: (xs ++ right)

appendAssoc :
  (a : List el) -> (b : List el) -> (c : List el) ->
    a ++ (b ++ c) = (a ++ b) ++ c
```

SEL4

- microkernel
- security enclave on iOS, among others
- no buffer overflows
- no null-pointer exceptions
- no use-after-free
- integrity
- confidentiality
- written in C
- verified with Haskell, Isabelle/HOL

COMPCERT

- C compiler
- verified with Coq
- output of register allocator checked by verified code

TOOLS

- Isabelle/HOL
- Coq
- Agda
- Idris
- ATS
- ACL2

USEFUL PROPERTIES

- commutativity $a \circ b = b \circ a$
- reflexivity $a :: a$
- symmetry $a :: b \Rightarrow b :: a$
- antisymmetry $a :: b, b :: a \Rightarrow a = b$
- transitivity $a :: b, b :: c \Rightarrow a :: c$

FANCY PROPERTIES

```
interface Functor (f : Type -> Type) where
    map : (func : a -> b) -> f a -> f b
```

FUNCTOR LAWS

```
interface Functor f => VerifiedFunctor (f : Type -> Type) where

  functorIdentity : {a : Type} -> (g : a -> a) ->
                    ((v : a) -> g v = v) -> (x : f a) ->
                      map g x = x

  functorComposition : {a : Type} -> {b : Type} -> (x : f a) ->
                       (g1 : a -> b) -> (g2 : b -> c) ->
                         map (g2 . g1) x = (map g2 . map g1) x
```

ANIMALS

```
data Animal : Type where
  Dillo : Liveness -> Weight -> Animal
  Parrot : String -> Weight -> Animal

runOverAnimal : Animal -> Animal
runOverAnimal (Dillo liveness weight) = Dillo Dead weight
runOverAnimal (Parrot sentence weight) = Parrot "" weight
```

ANIMAL FUNCTOR

```
data Animal : Type -> Type where
  Dillo : Liveness -> weight -> Animal weight
  Parrot : String -> weight -> Animal weight

runOverAnimal : Animal weight -> Animal weight
runOverAnimal (Dillo liveness weight) = Dillo Dead weight
runOverAnimal (Parrot sentence weight) = Parrot "" weight

implementation Functor Animal where
  map f (Dillo liveness weight) = Dillo liveness (f weight)
  map f (Parrot sentence weight) = Parrot sentence (f weight)
```

CONCLUSION

- find the combinator
- make it a monoid
- write properties
- test properties
- prove properties
- find the functor
- watch your brain grow
- sleep soundly