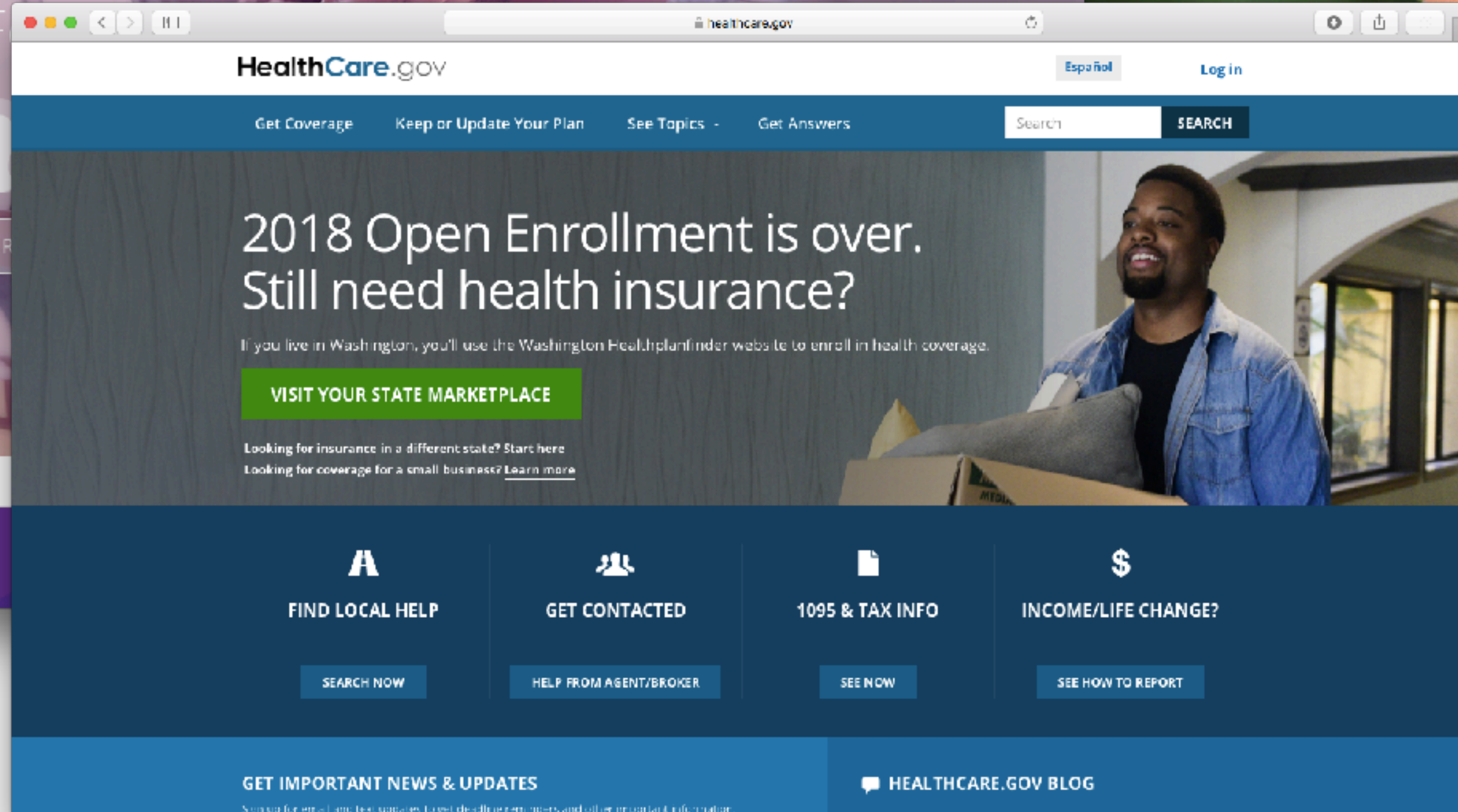
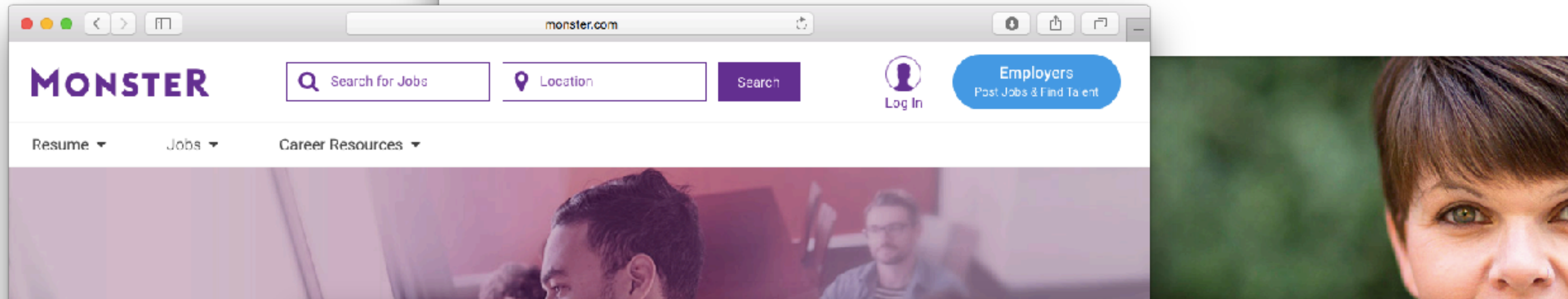
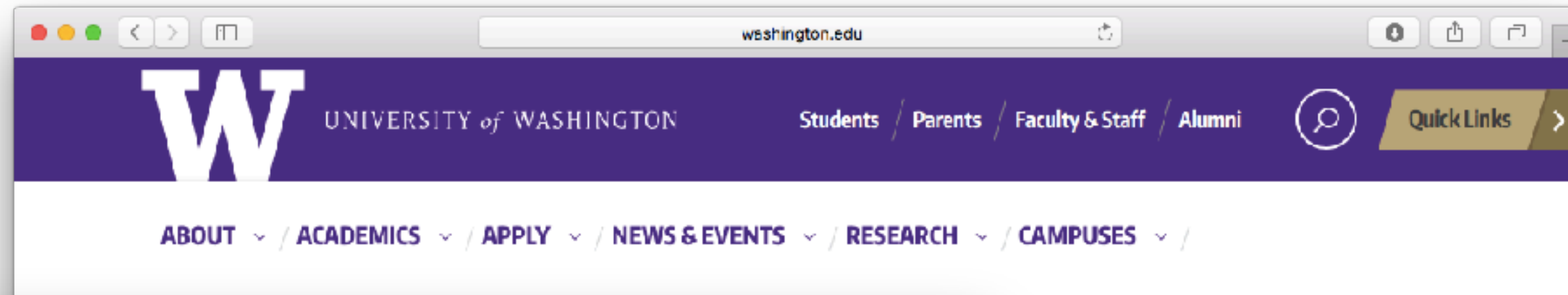
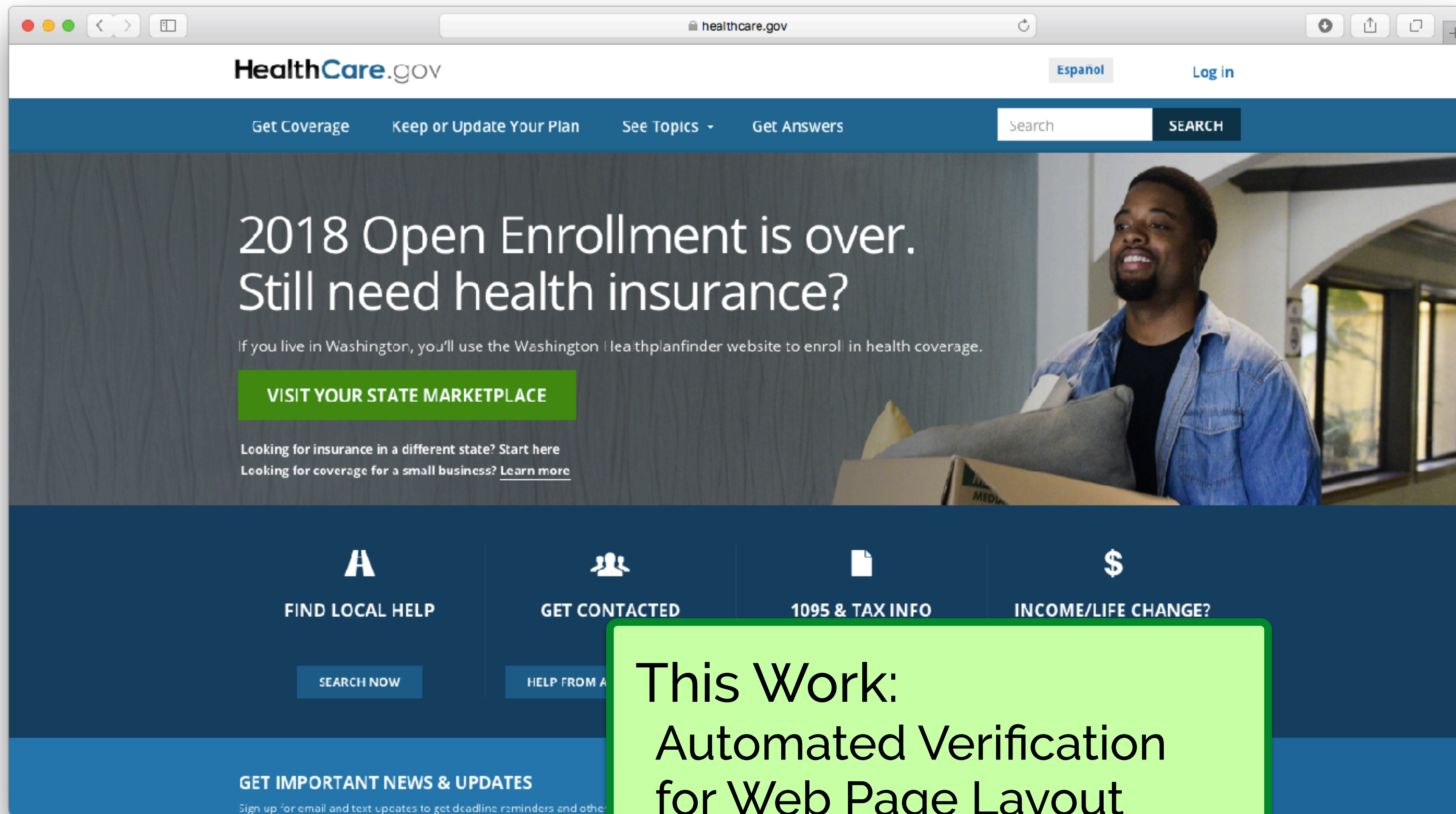


# Verifying That Web Pages Have Accessible Layout

**Pavel Panchekha**, Adam T. Geller, Michael D. Ernst,  
Zachary Tatlock, Shoaib Kamil







3.6M lines



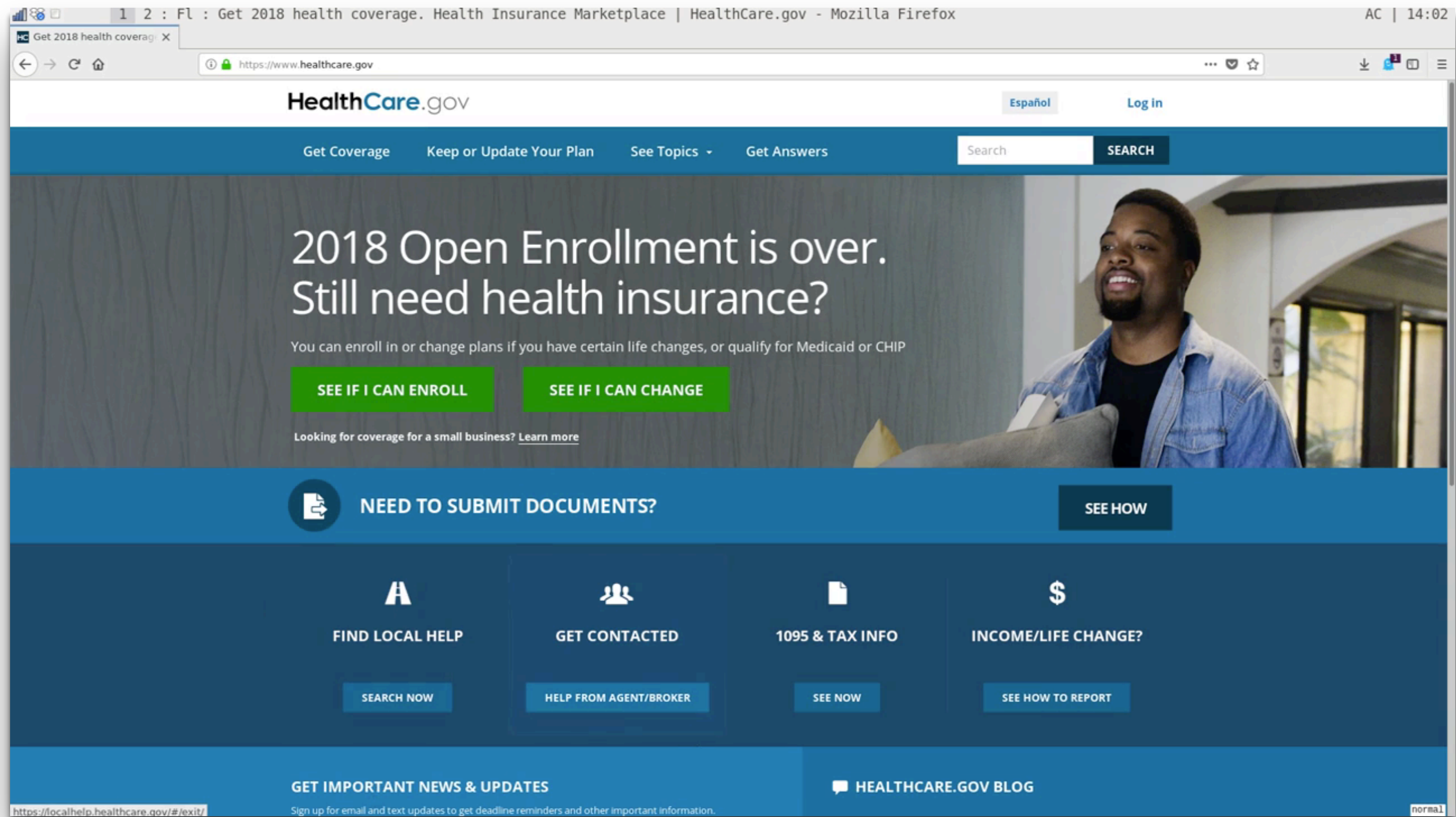
Backend

Layout

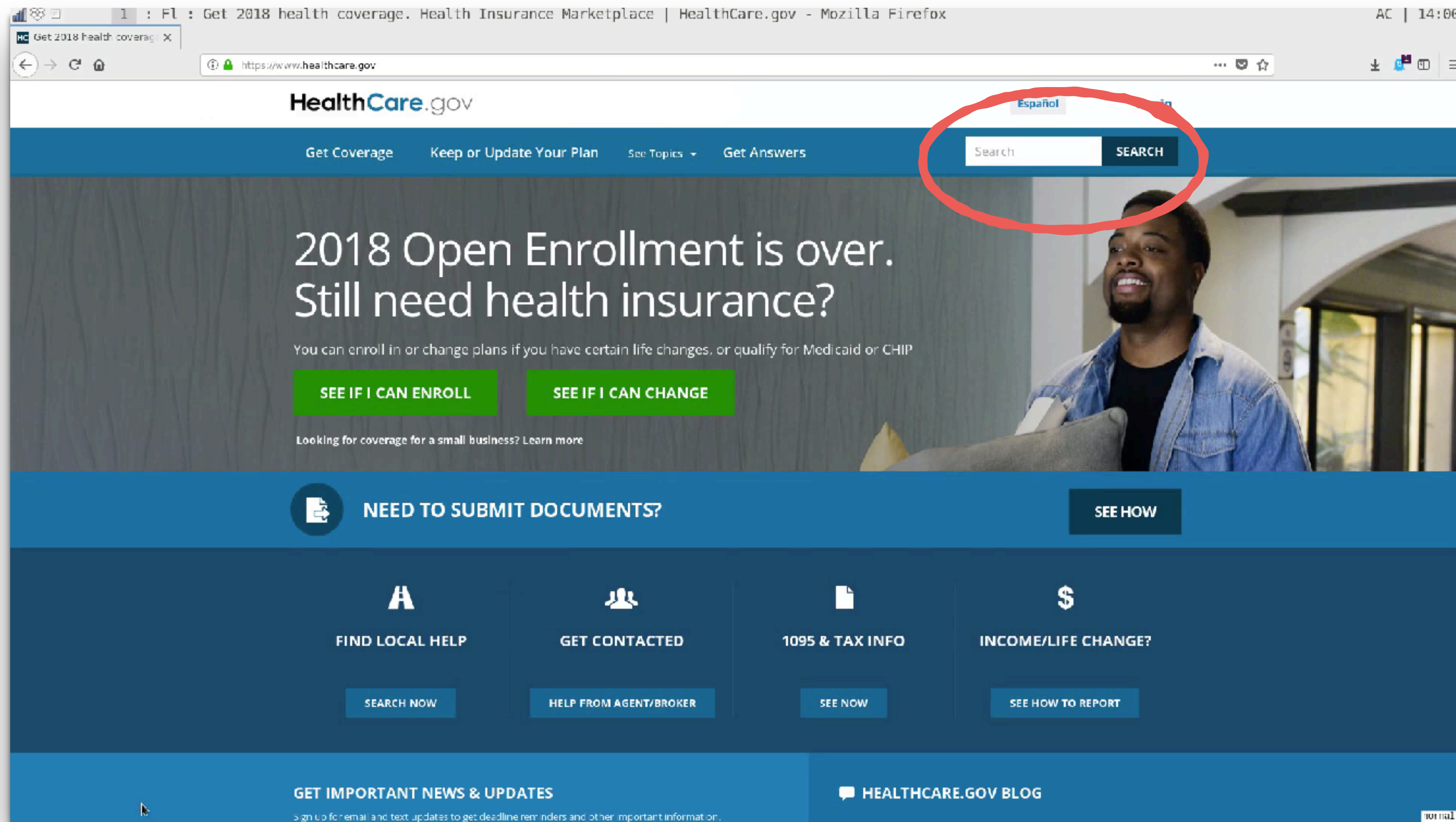
Script

Config

*Existing verification work*



Multiple layouts, responsive design



Usability, accessibility, mobile-friendliness

*ADA Compliance*

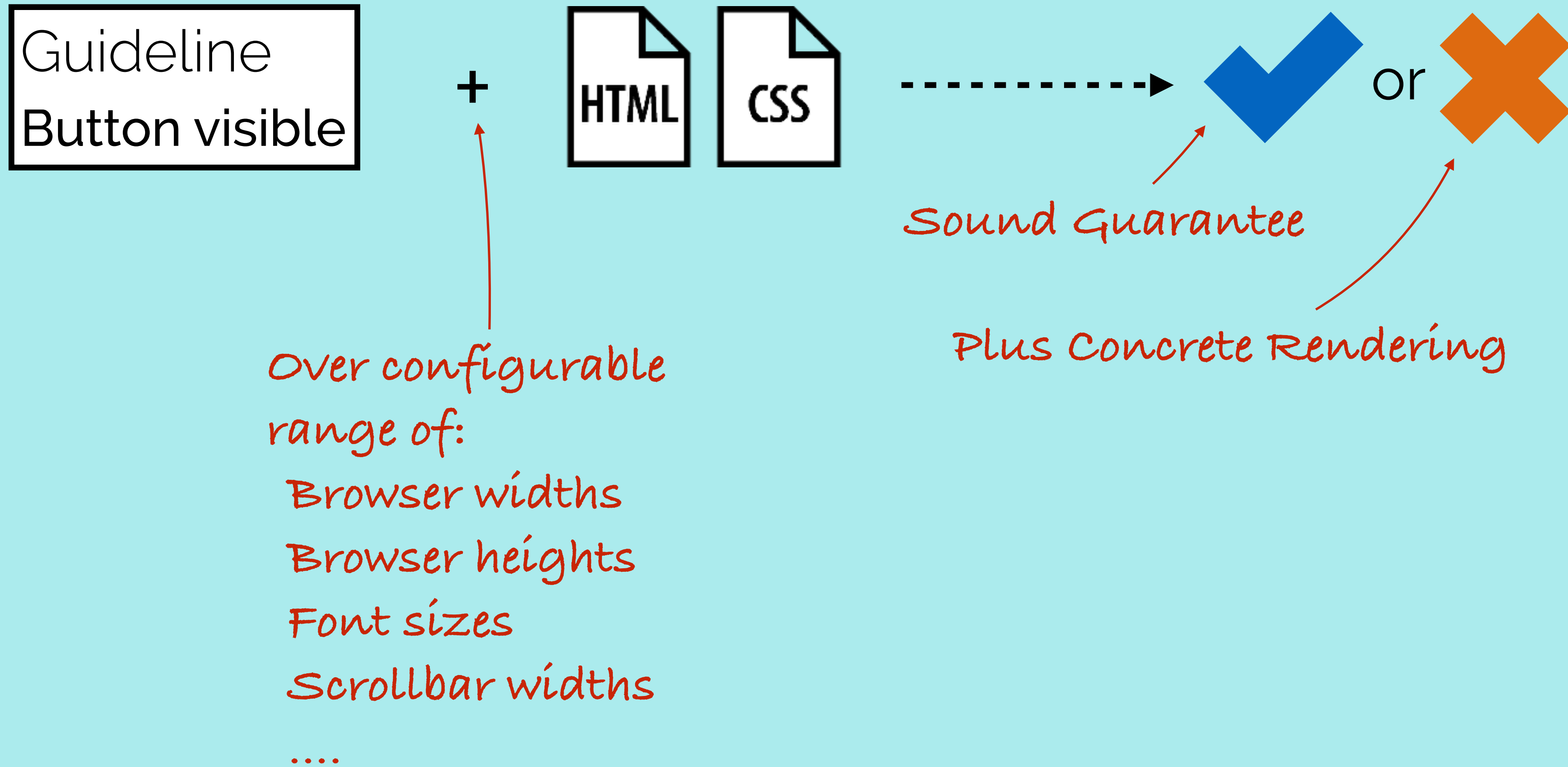
Complex Behavior + Correctness Properties

**HELP  
WANTED**

VizAssert

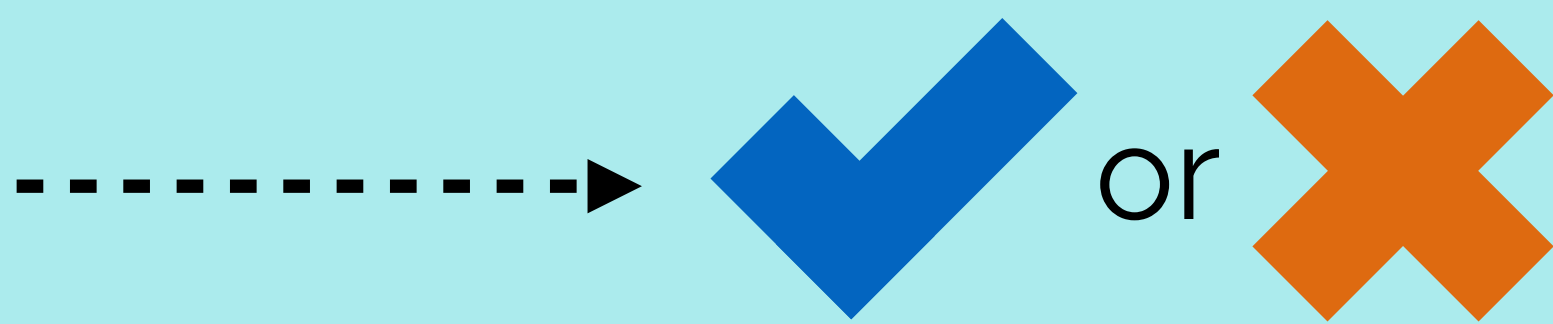
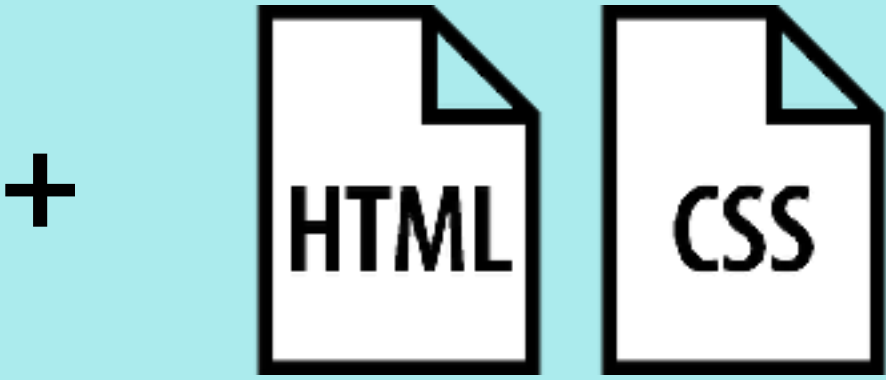
Automatic Verification  
for Web Page Layout

# VizAssert



# VizAssert

Guideline  
Button visible

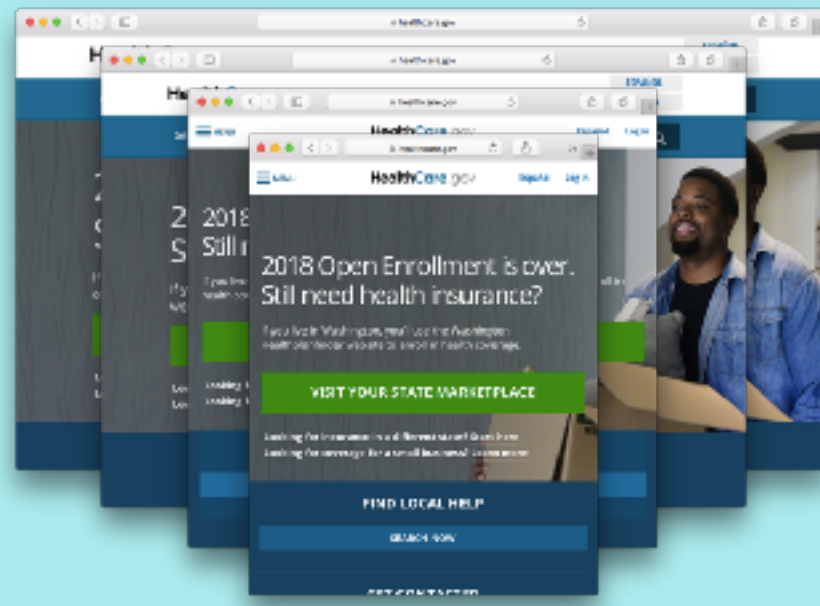


1.

$\forall b_1, b_2 \in \mathcal{B},$   
 $b_1 \in \$(.search-button) \wedge$   
 $b_2 \in \$(.toolbar) \implies$   
 $within(b_1, b_2)$

Visual Logic

2.



Renderings

3.

```
(set-logic QF_LRA)
(declare-type Box ...)
(declare-const b1 Box)
(declare-const b2 Box)
(assert (not ...))
```

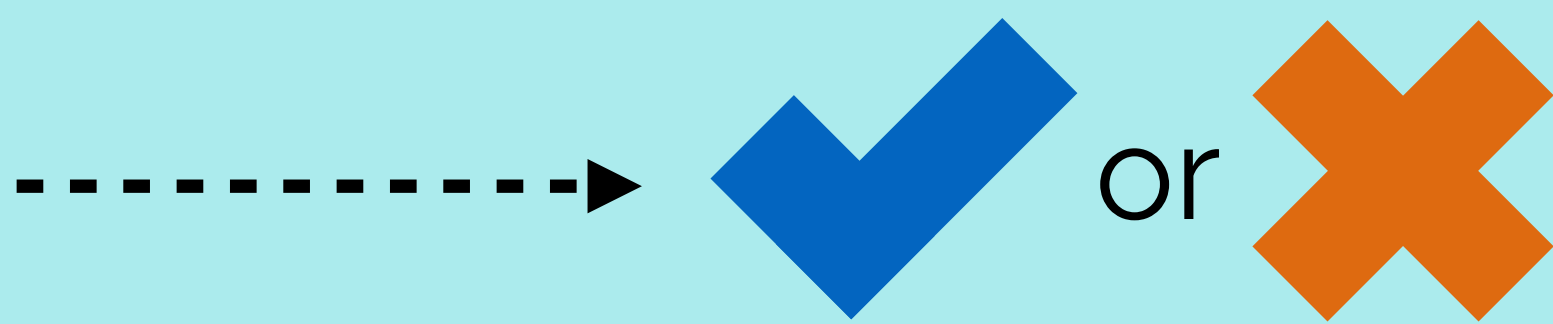
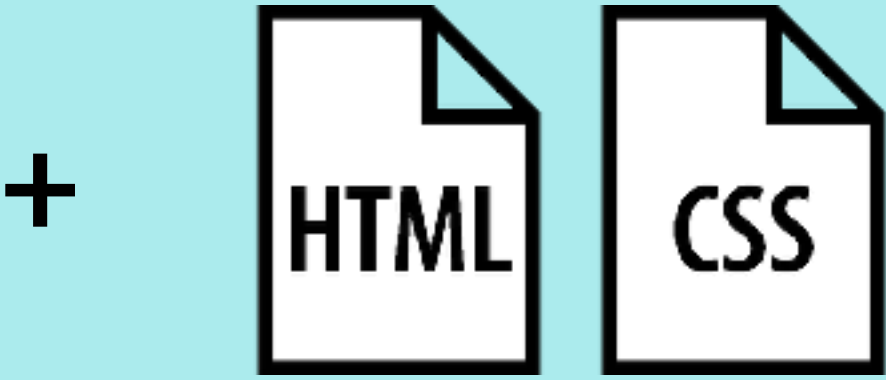
SMT Query

Z3



# VizAssert

Guideline  
Button visible

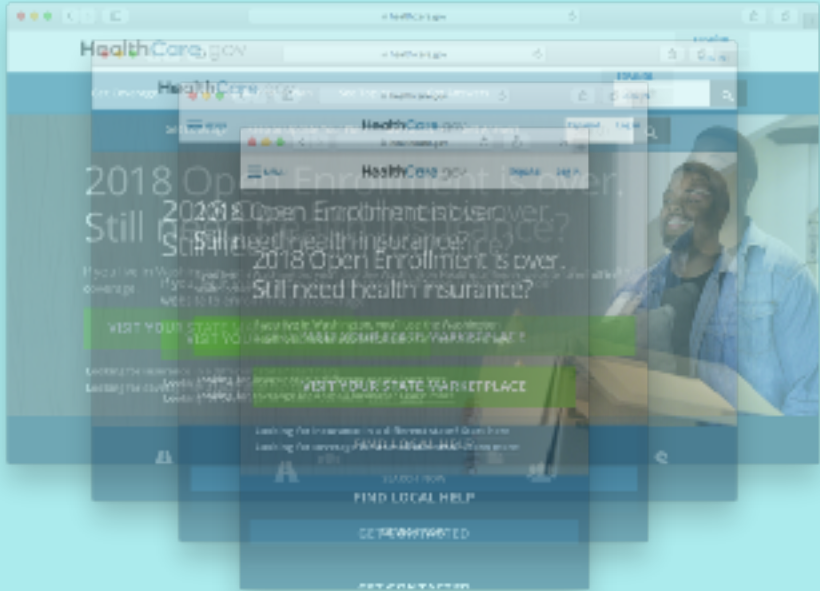


1.

$\forall b_1, b_2 \in \mathcal{B},$   
 $b_1 \in \$(.search-button) \wedge$   
 $b_2 \in \$(.toolbar) \implies$   
 $\text{within}(b_1, b_2)$

Visual Logic

2.



Renderings

3.

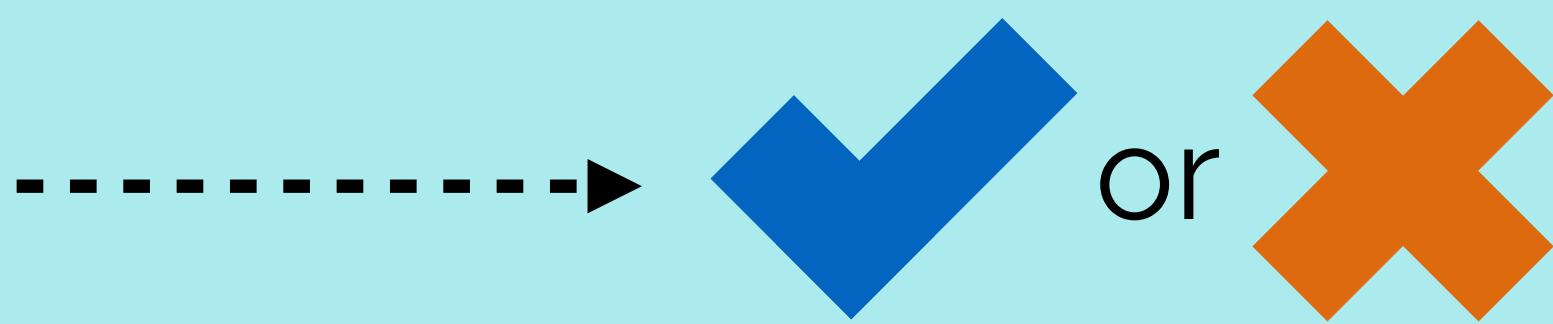
(set-logic QF\_LRA)  
(declare-type Box ...)  
(declare-const b1 Box)  
(declare-const b2 Box)  
(assert (not ...))

SMT Query

Z3

# VizAssert

Guideline  
Button visible

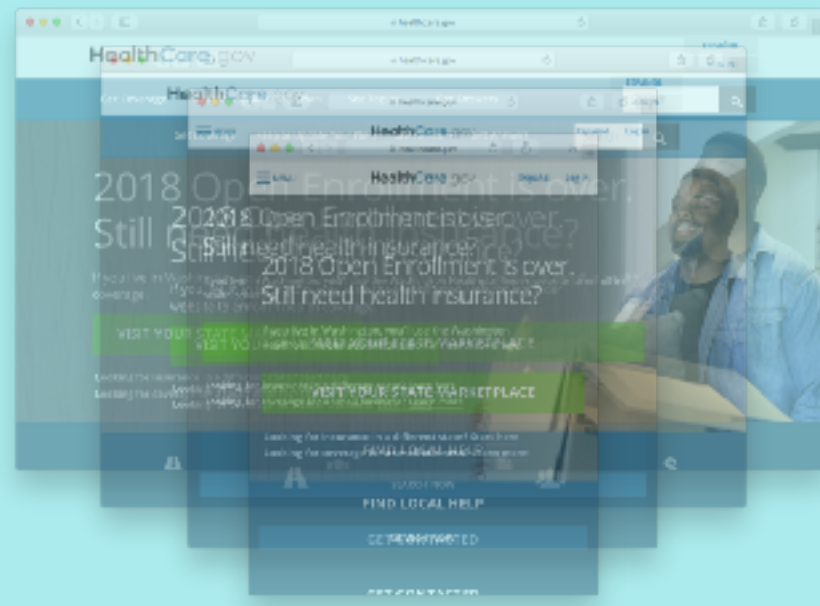


1.

$\forall b_1, b_2 \in \mathcal{B},$   
 $b_1 \in \$(.search-button) \wedge$   
 $b_2 \in \$(.toolbar) \implies$   
 $\text{within}(b_1, b_2)$

Visual Logic

2.



Renderings

3.

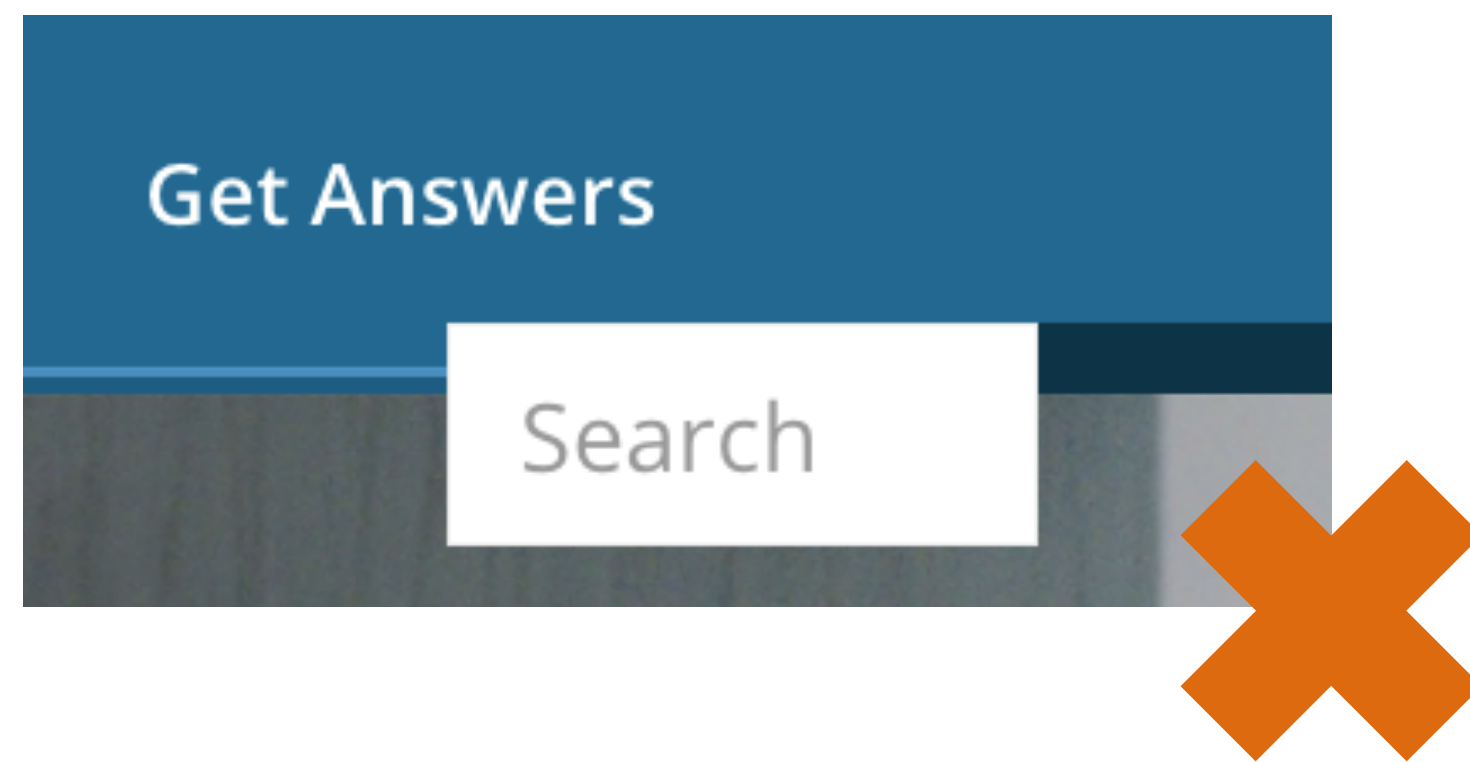
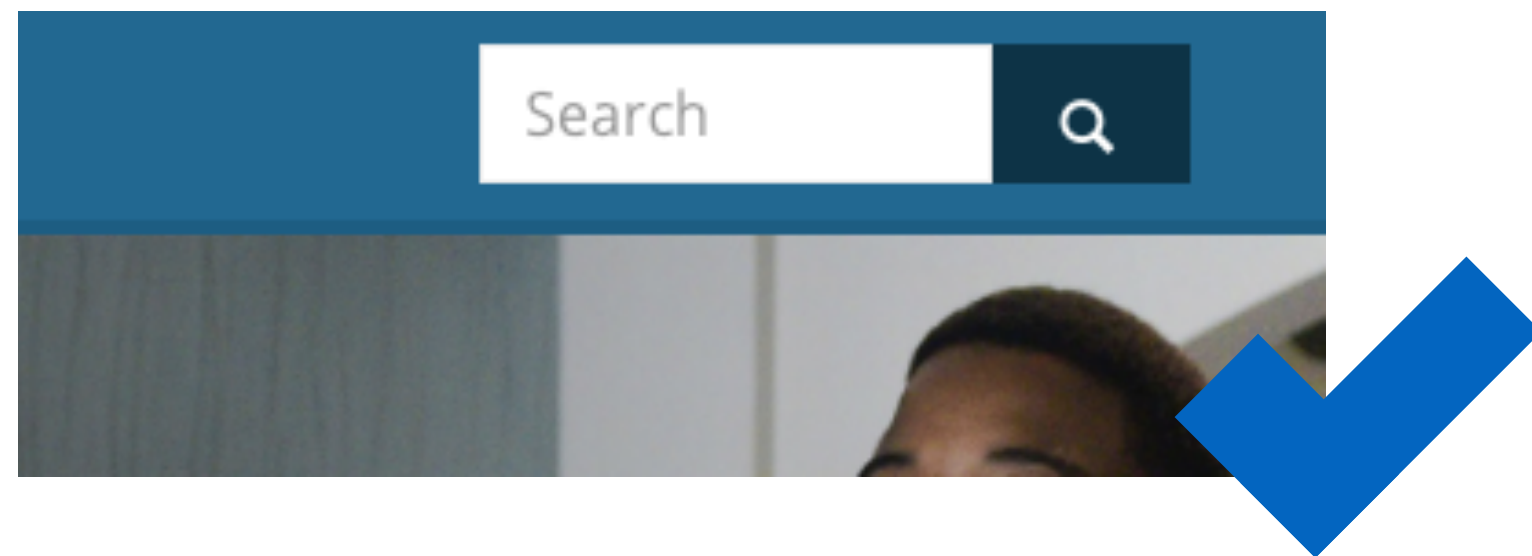
(set-logic QF\_LRA)  
(declare-type Box ...)  
(declare-const b1 Box)  
(declare-const b2 Box)  
(assert (not ...))

SMT Query

Z3

Guideline

Button visible



Visual Logic

$\forall b_1, b_2 \in \mathcal{B},$  *universal quantifier*

$b_1 \in \$(.search-button) \wedge$

$b_2 \in \$(.toolbar) \implies$

$\text{within}(b_1, b_2)$

*Structural properties*

*Layout property*

universal  
quantifier

$\forall b_1, b_2, \dots$

HTML *Structural  
properties*

root

$b.parent$

$b.previous$

$b.first-child$

$b \in \$(selector)$

...



Visual Logic

CSS *Layout  
property*

$b.left$

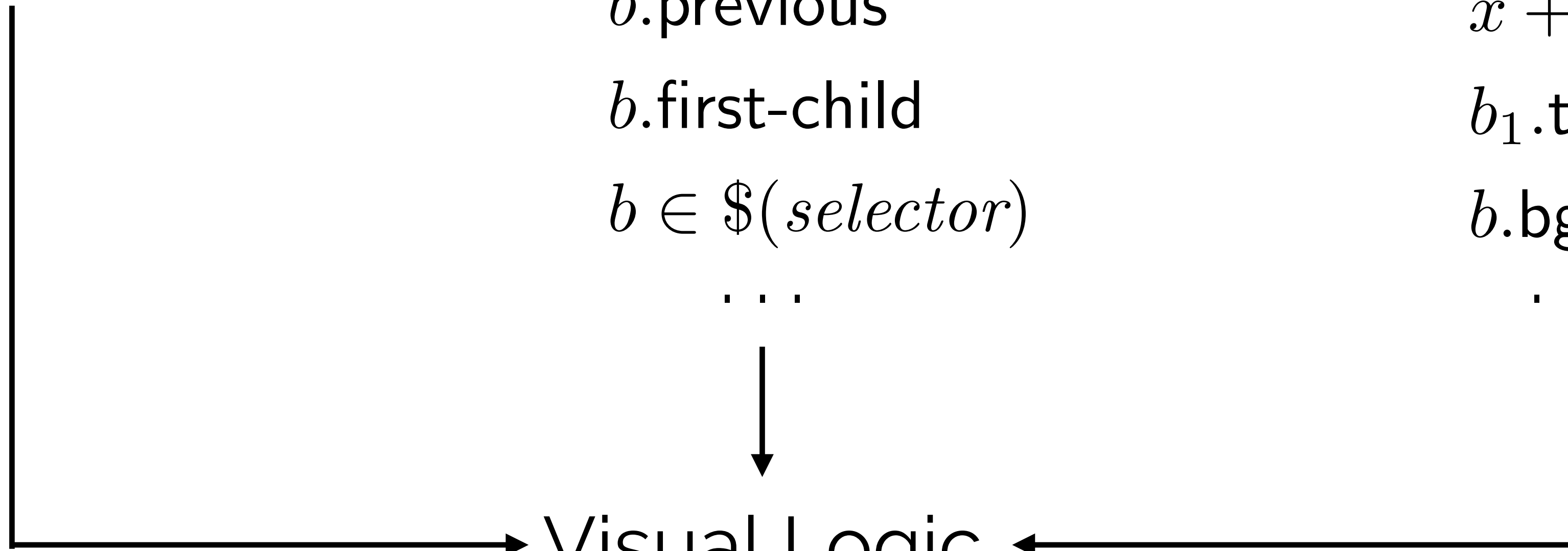
$b.top[margin]$

$x + 2y - z$

$b_1.top < b_2.top$

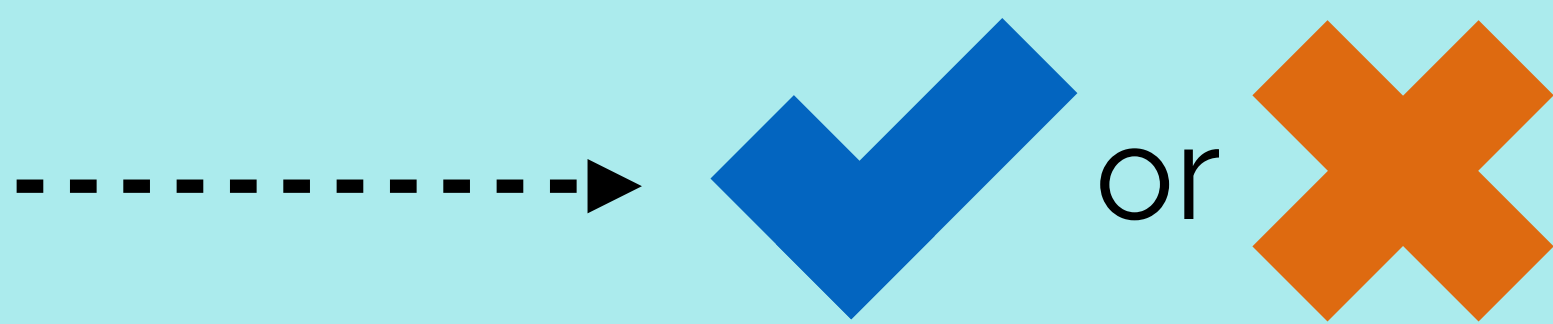
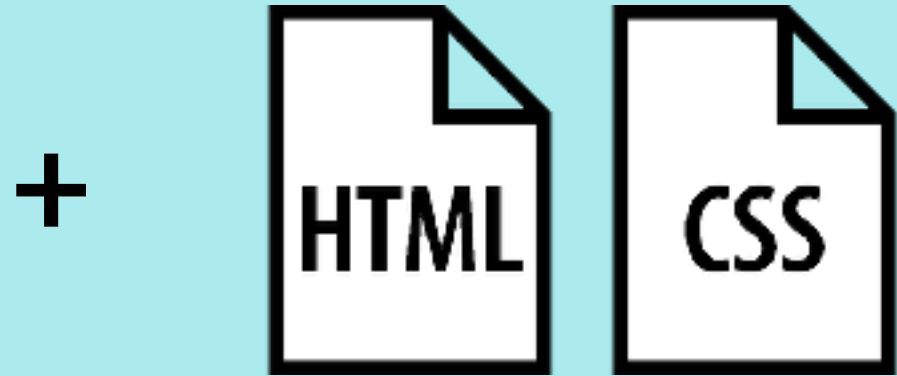
$b.bg.r$

...



# VizAssert

Guideline  
Button visible

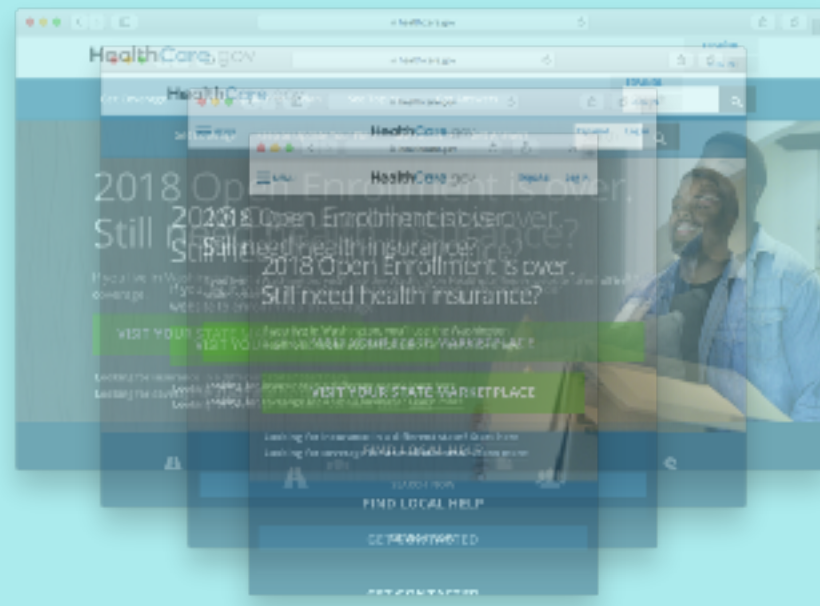


1.

$\forall b_1, b_2 \in \mathcal{B},$   
 $b_1 \in \$(.search-button) \wedge$   
 $b_2 \in \$(.toolbar) \implies$   
 $\text{within}(b_1, b_2)$

Visual Logic

2.



Z3

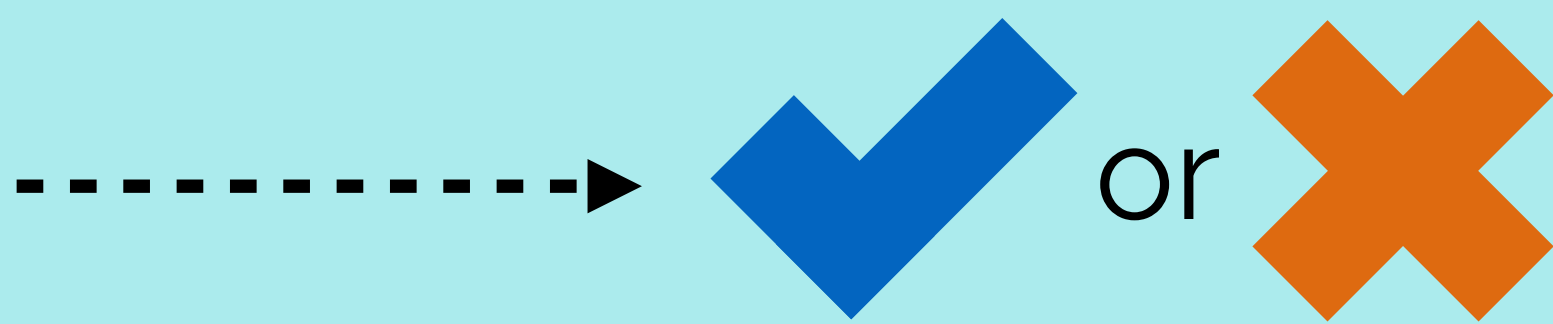
3.

(set-logic QF\_LRA)  
(declare-type Box ...)  
(declare-const b1 Box)  
(declare-const b2 Box)  
(assert (not ...))

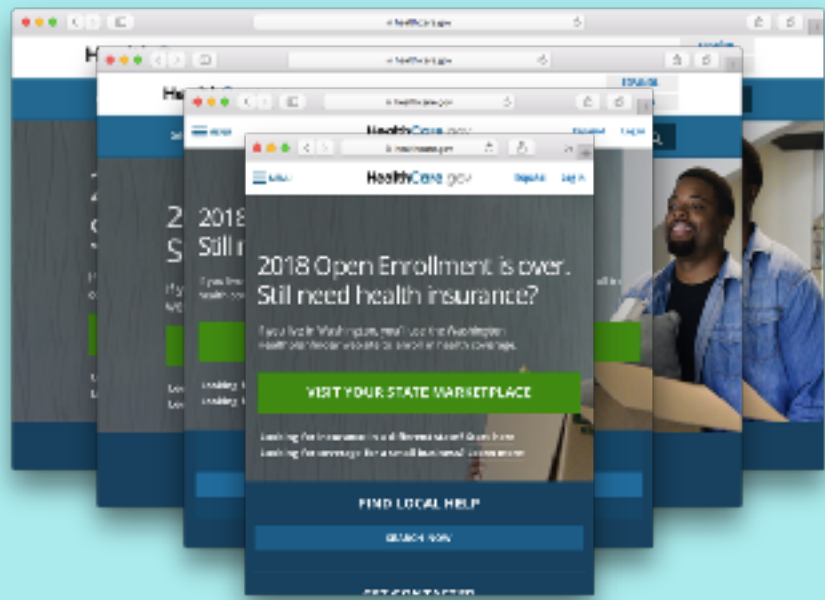
SMT Query

# VizAssert

Guideline  
Button visible



2.



Renderings

1.

$\forall b_1, b_2 \in \mathcal{B},$   
 $b_1 \in \$(.search-button) \wedge$   
 $b_2 \in \$(.toolbar) \implies$   
 $\text{within}(b_1, b_2)$

Visual Logic

Z3

3.

```
(set-logic QF_LRA)
(declare-type Box ...)
(declare-const b1 Box)
(declare-const b2 Box)
(assert (not ...))
```

SMT Query

```
A font-feature-settings (descriptor) P
:act/
addi background-size grid-column-end <ratio>
::aft <basic-shape> grid-column-end :read-only
align ::be border-inline-start-color inline-size scroll-snap-coordinate
align <ble border margin-right text-decoration-color
align bloc borde @charset margin-top text-decoration-line
all blur borde @namespace U
<angl borde borde dpcm negative unicode-bidi
anima borde borde clea dpi :not() unicode-range
anima borde borde clip dppx :nth-child() :unresolved
anima borde borde cm drop-shadow() :nth-last-child() unset
anima borde borde colo E :nth-last-of-type() <url>
anima borde borde <col element() :nth-of-type() url()
anima borde borde colu ellipse() <number> user-zoom
anima borde borde colu em O V
anima borde borde colu :empty object-fit :valid
anima borde borde colu empty-cells object-position var()
@anne borde borde colu :enabled offset-block-end vertical-align
annot borde borde colu ex offset-block-start vh
attr borde borde colu F offset-inline-end @viewport
B borde borde colu fallback offset-inline-start visibility
::bac borde borde colu filter :only-child :visited
back: borde borde colu :first :only-of-type vmax
back: borde borde colu :first-child opacity vmin
back: borde botto cont ::first-letter (:first-letter) vw
back: borde box-c cont ::first-line (:first-line) W
back: borde box-s <cou :first-of-type white-space
back: borde box-s coun flex orientation widows
back: borde brea @cou flex-basis @ornaments width
back: borde brea @cou flex-direction ornaments() width
back: borde brea cubi flex-flow orphans will-change
back: borde brig curs flex-grow :out-of-range word-break
back: borde C <cus flex-shrink outline word-spacing
borde calc D flex-wrap outline-color word-wrap
borde capt: :def float outline-offset writing-mode
ch deg :focus outline-style XYZ
@cha :dir font outline-width z-index
dire font overflow zoom
char: @font-face overflow-wrap Others
:dis font-family (property) overflow-x --*
font-family (descriptor) overflow-y
```

102 standards

100k+ words

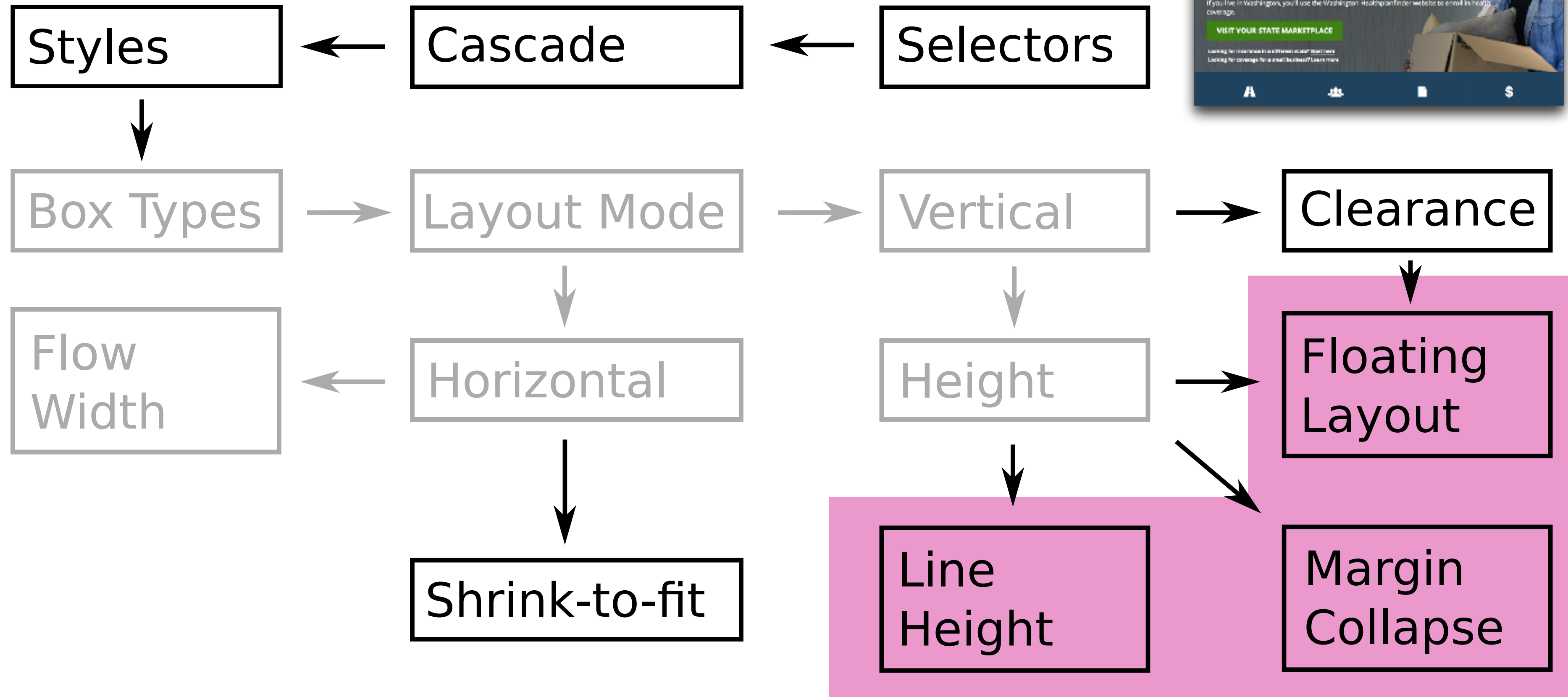
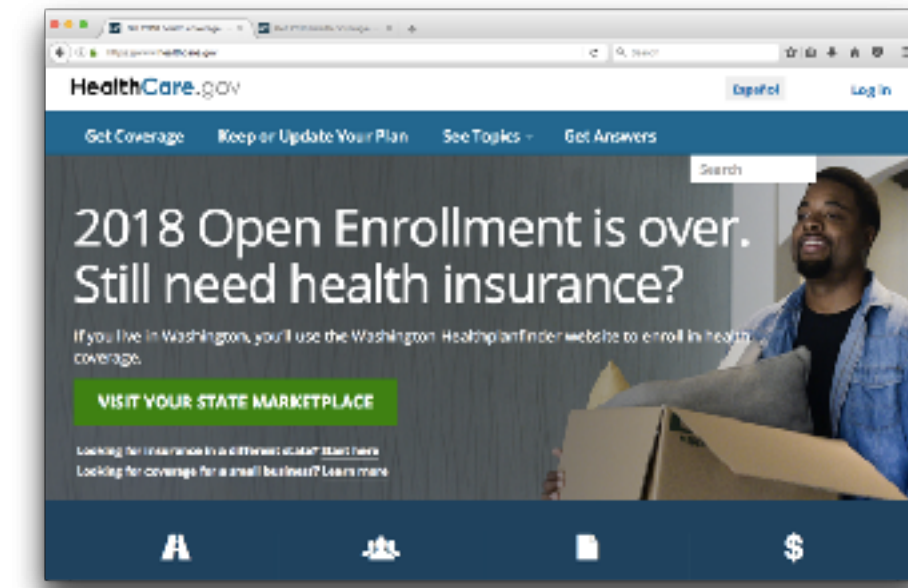
487 properties



**Symbolic**  
web browser  
**in SMT**

# Formalized CSS Features

The CSS People '16





# Finitization Reductions

$\forall \exists \forall \exists$

Quantifier alternation  
Line Height

$\mu x.P(x)$

Fixpoints, reachability  
Margin Collapsing

$\text{sort}(L)$

Data structures  
Floating Layout



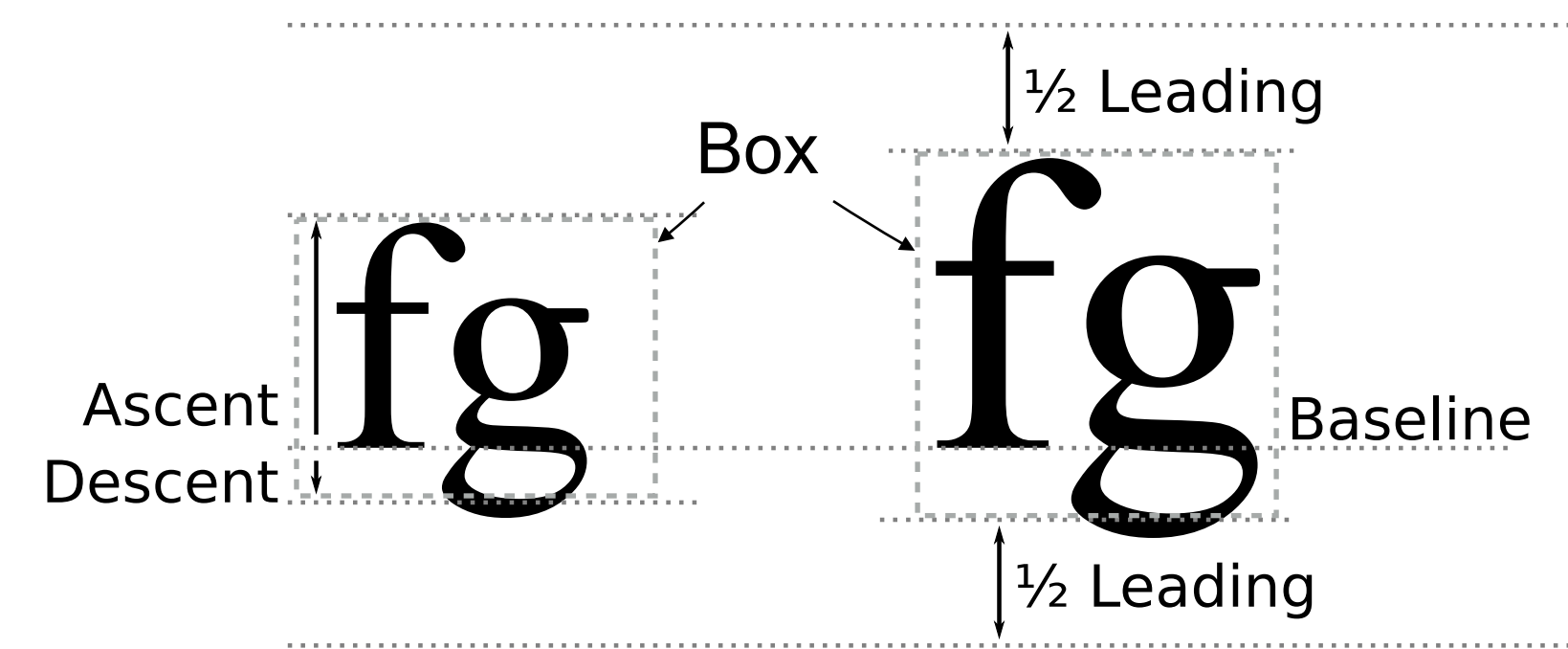
Finite number of registers

Avoid the jargon in audio  
Faster

# Line Height

“The line box height is the distance between the **uppermost** box top and the **lowermost** box bottom.”

*Maximum  
over all boxes*



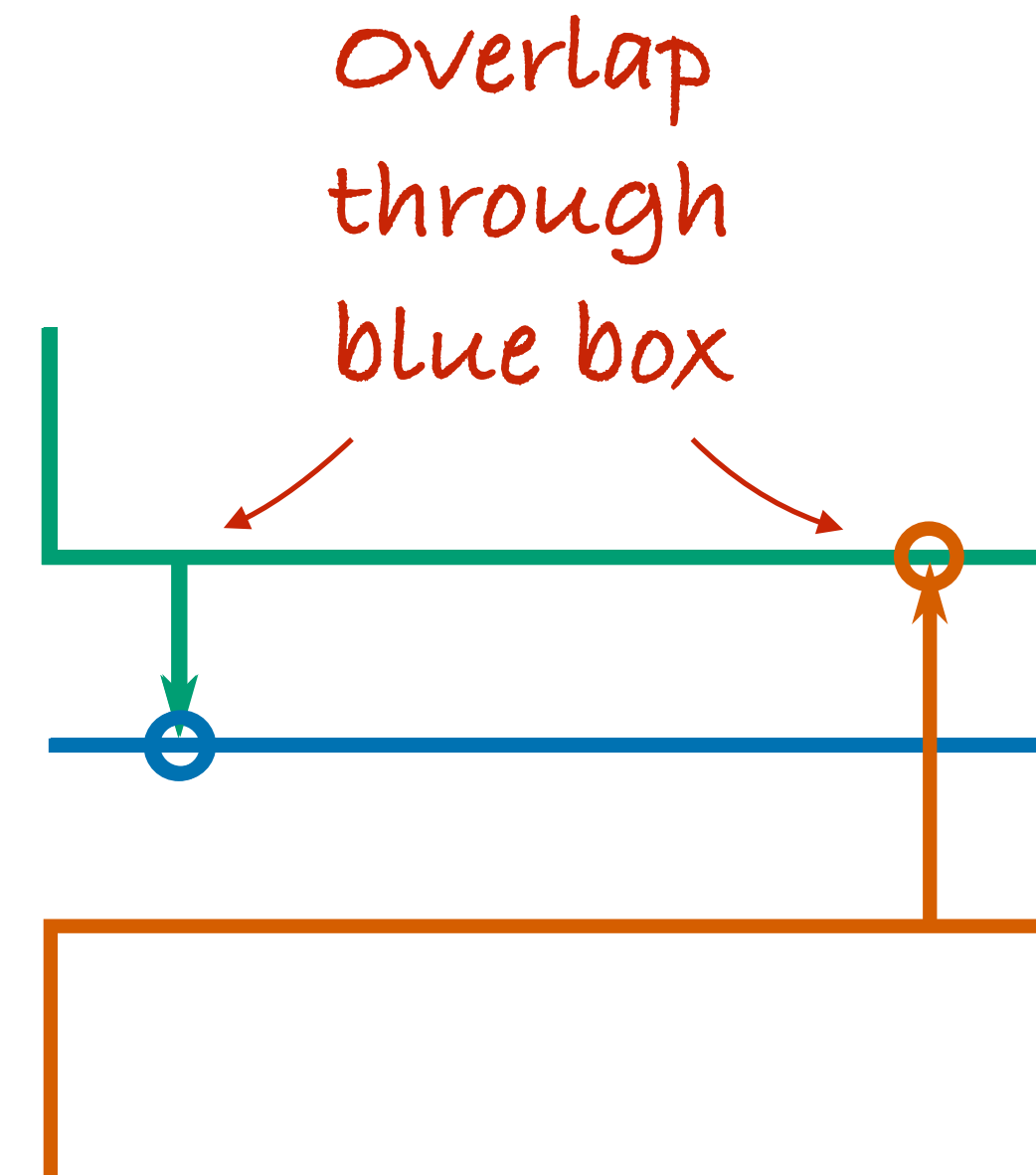
Reduced to 2-register encoding

Incremental maximum/minimum computations

# Margin collapsing

*Reachability  
condition*

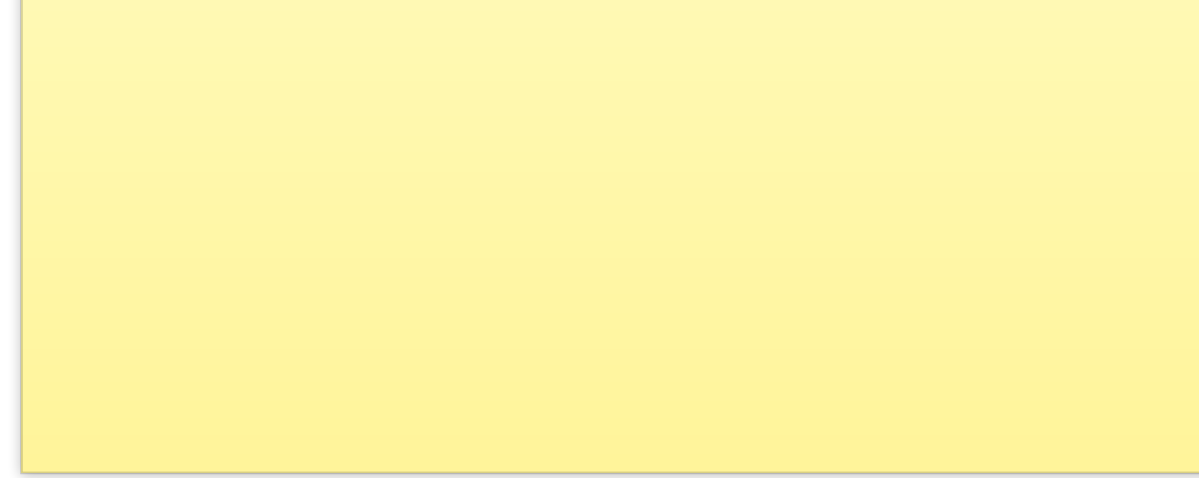
“**Adjoining** vertical margins collapse; two margins are adjoining if and only if: ...”



Reduced to 6-register encoding

Collapse margins box-by-box as they're laid out

# Float Semantics

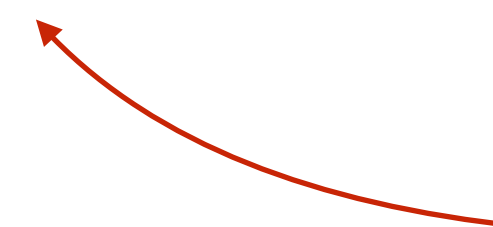


Right floats  
analogous



“Here are the precise rules that govern the behavior of left floats:

1. A floating box's left outer edge may not be to the left of the left edge of its containing block.
2. A floating box must be to the right of the right outer edge, or below the bottom outer edge, of any earlier floating box.
3. The right outer edge of a left-floating box may not be to the right of the left outer edge of any right-floating box that is next to it.
4. A floating box's outer top may not be higher than the top of its containing block.
5. The outer top of a floating box may not be higher than the outer top of any block or floated box generated by an element earlier in the source document.
6. The outer top of an element's floating box may not be higher than the top of any line-box containing a box generated by an earlier element.
7. A left-floating box that has another left-floating box to its left may not have its right outer edge to the right of its containing block's right edge.
8. A floating box must be placed **as high as possible**.
9. A left-floating box must be put **as far to the left as possible**.”

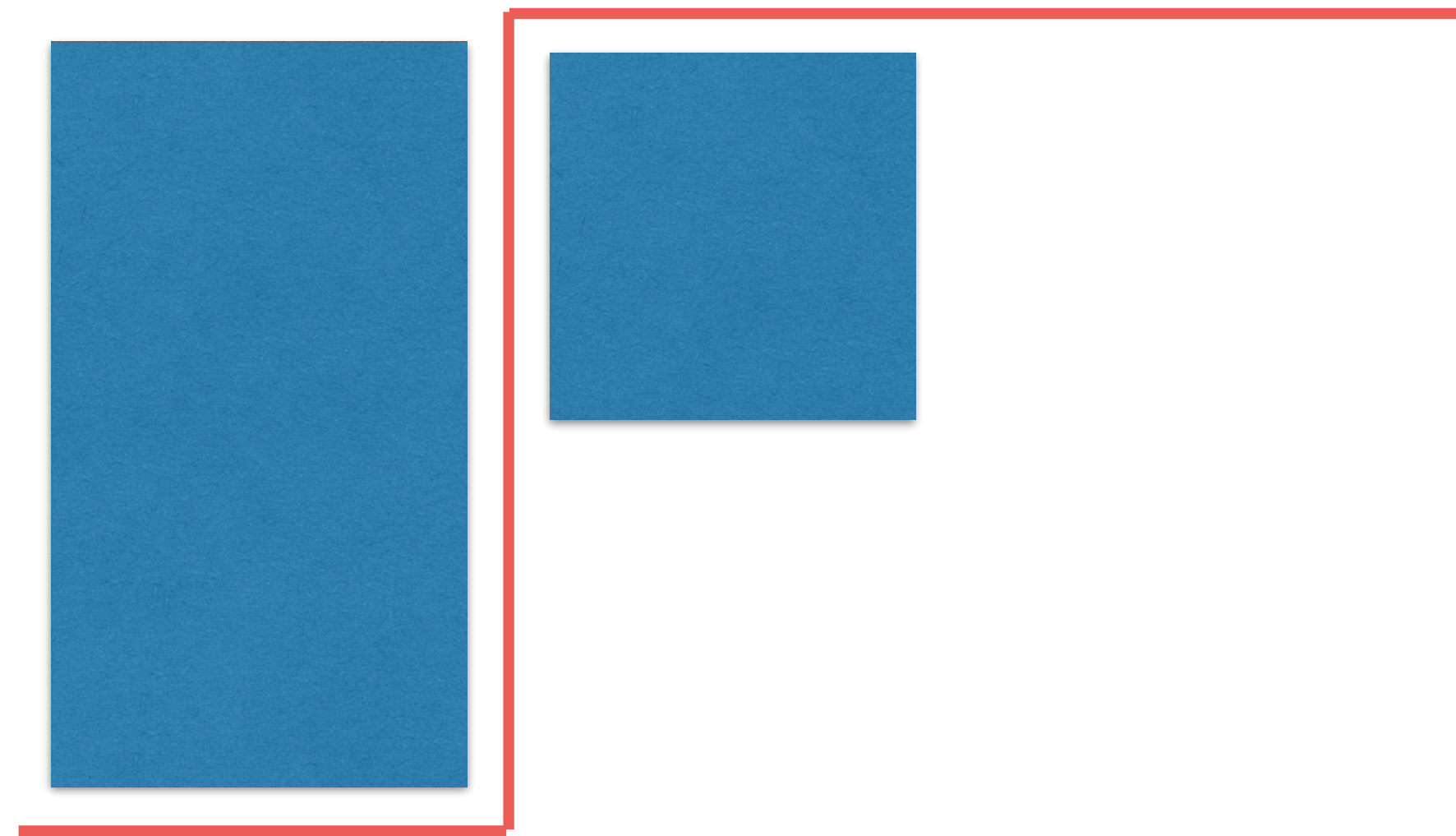


Optimization  
problem

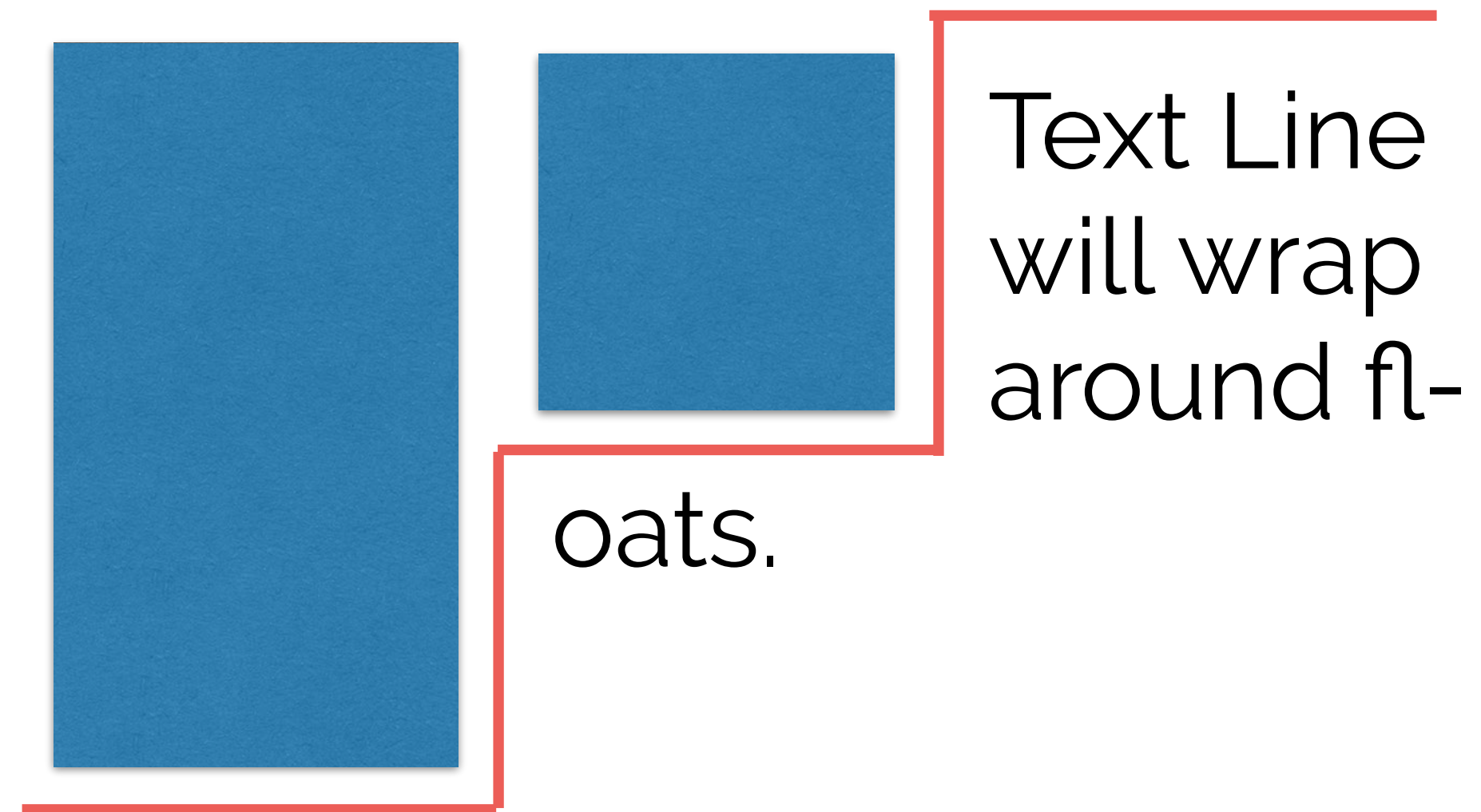
# Float Semantics



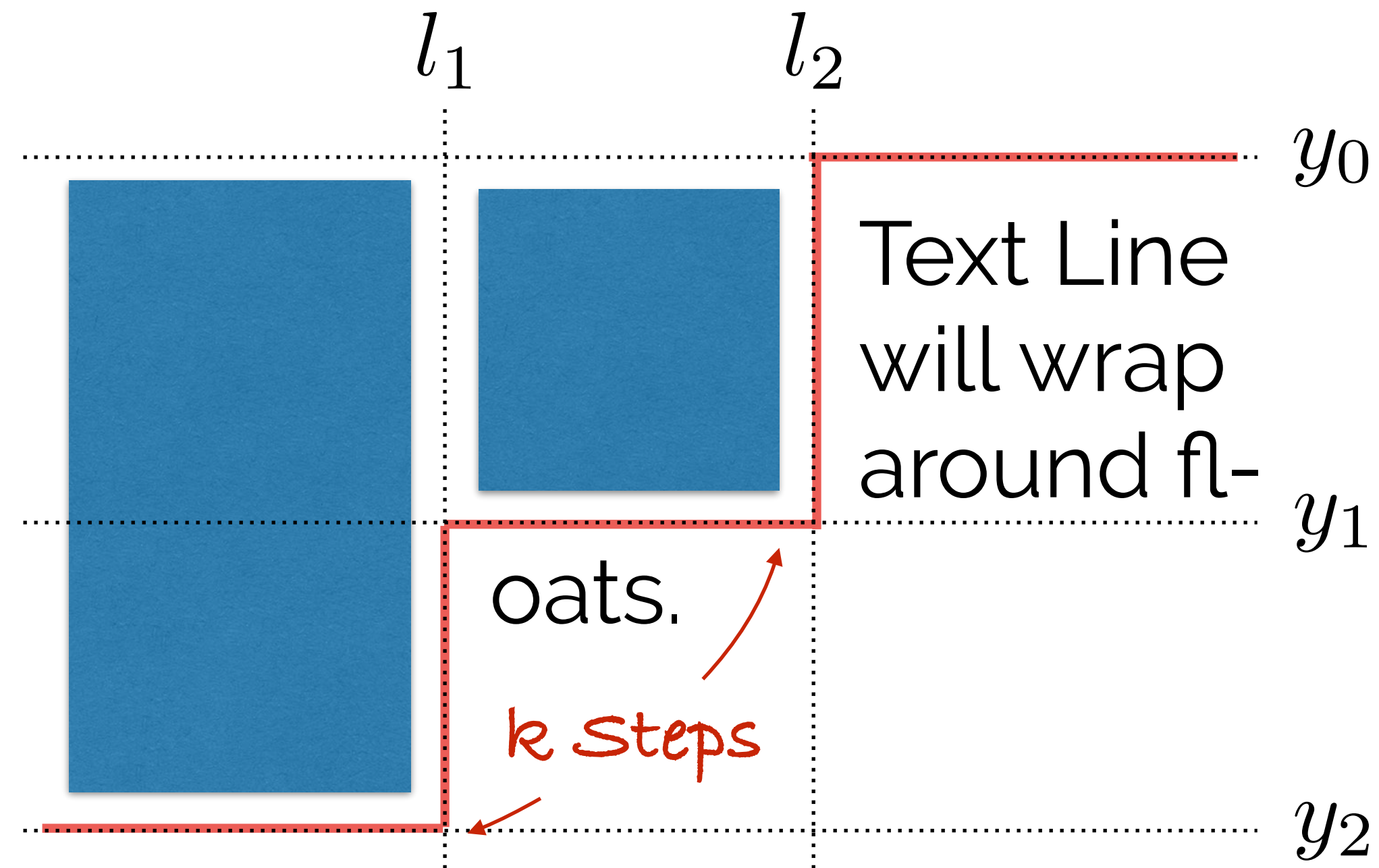
# Float Semantics



# Float Semantics



# Float Semantics



Reduced to  $(5k+2)$ -register encoding

In practice,  $k \leq 7$ ; search for sufficient value



# Finitization Reductions

$\forall \exists \forall \exists$

Quantifier alternation

Line Height

→ 2 registers

$\mu x.P(x)$

Fixpoints, reachability

Margin Collapsing

→ 5 registers

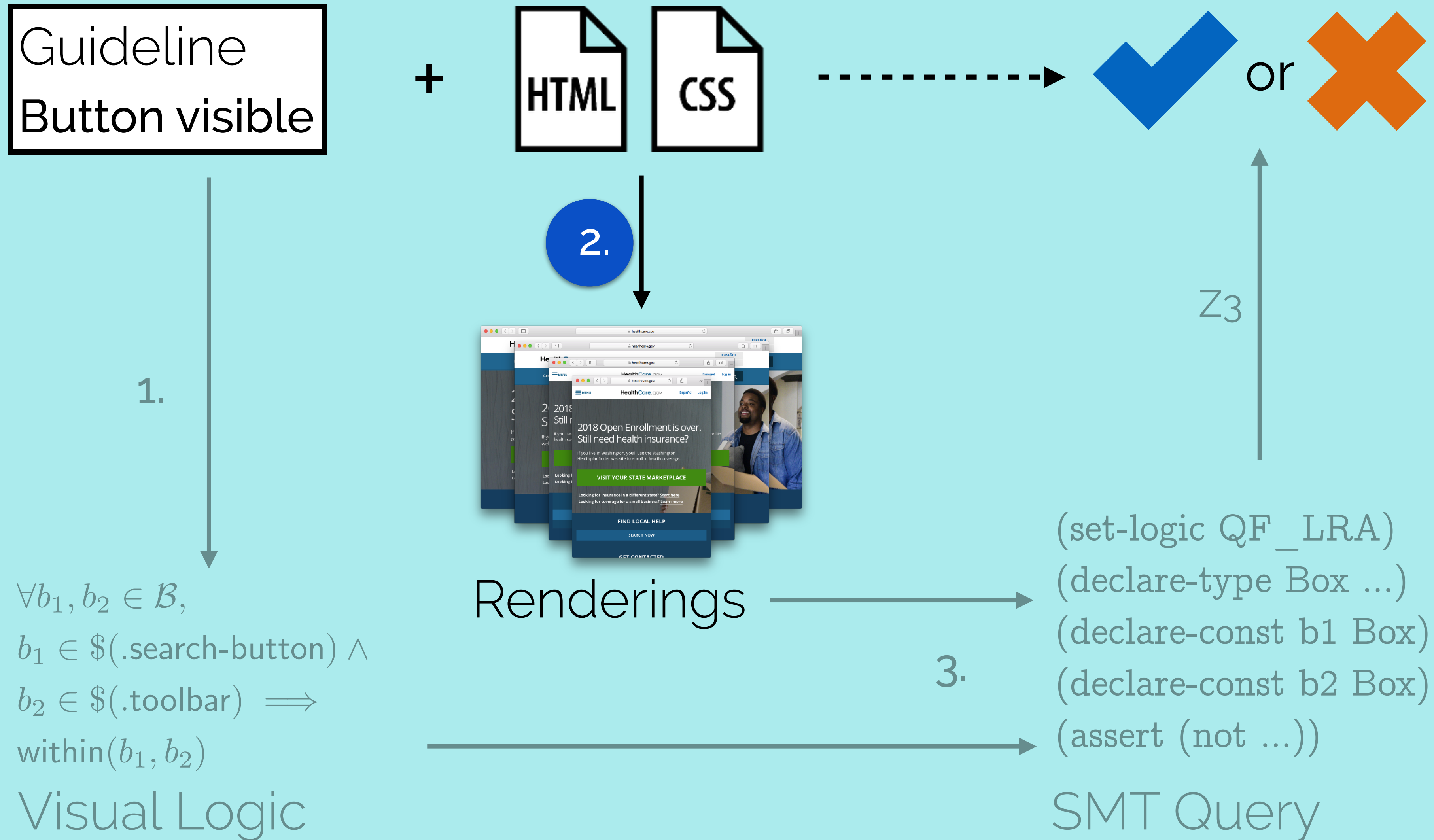
$\text{sort}(L)$

Data structures

Floating Layout

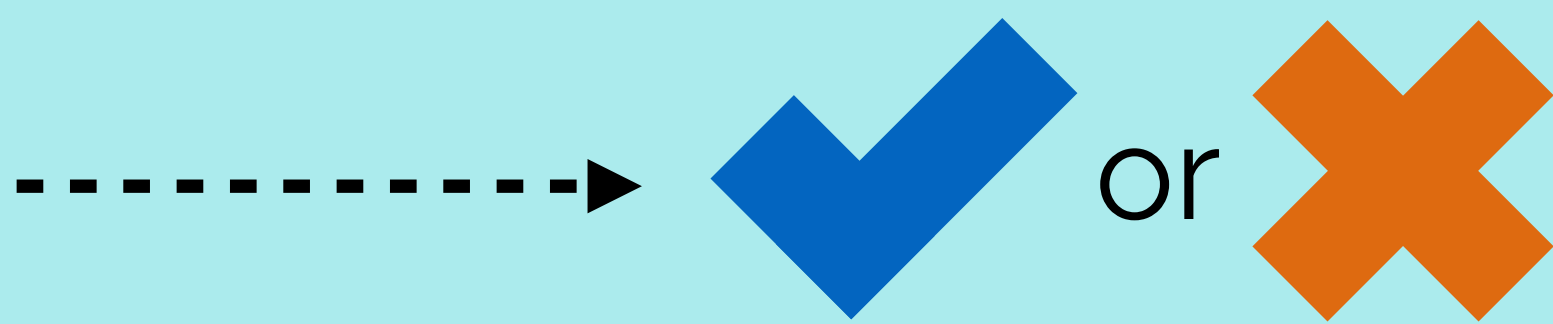
→  $5k+2$  registers

# VizAssert



# VizAssert

Guideline  
Button visible

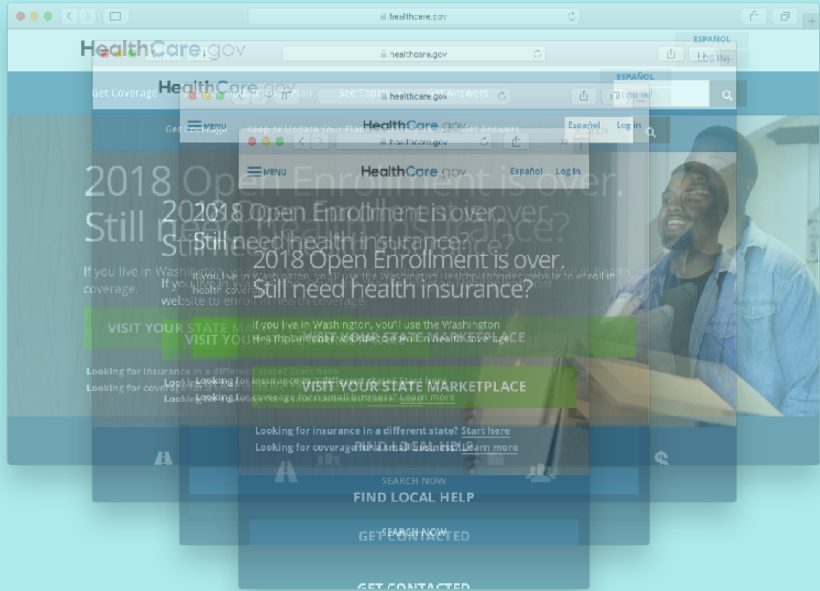


1.

$\forall b_1, b_2 \in \mathcal{B},$   
 $b_1 \in \$(.search-button) \wedge$   
 $b_2 \in \$(.toolbar) \implies$   
 $\text{within}(b_1, b_2)$

Visual Logic

2.



Renderings

3.

```
(set-logic QF_LRA)
(declare-type Box ...)
(declare-const b1 Box)
(declare-const b2 Box)
(assert (not ...))
```

SMT Query

Z3

# SMT Query

val *html* = "<html><body>...</body></html>"

val *css* = "#main { ... }; p li { ... }; ..."

abstract *params* : BrowserParams

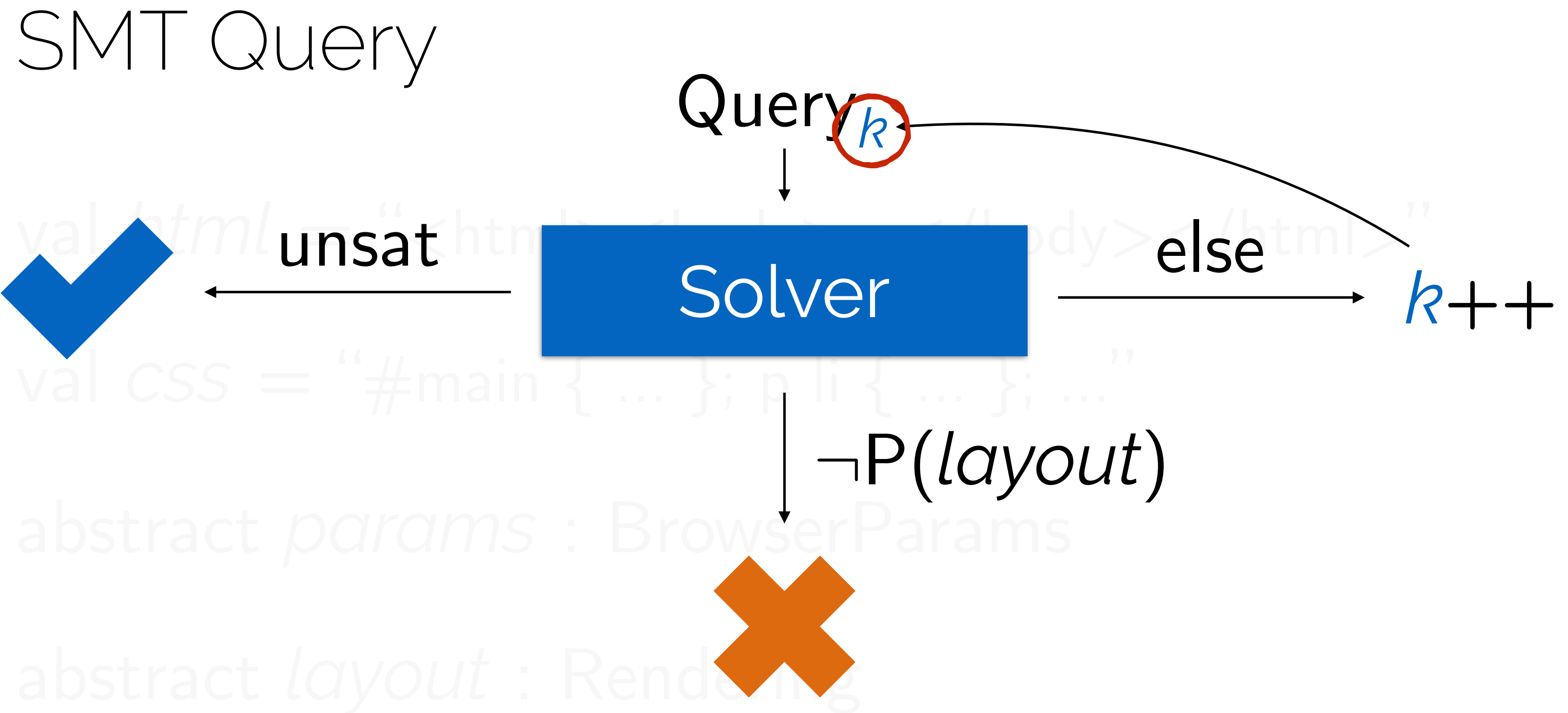
abstract *layout* : Rendering

require *layout* = render (*html*, *css*, *params*)

require  $\neg P(\textit{layout})$

user constraints, ex.  
browser width > 600

valid counterexample  
to assertion

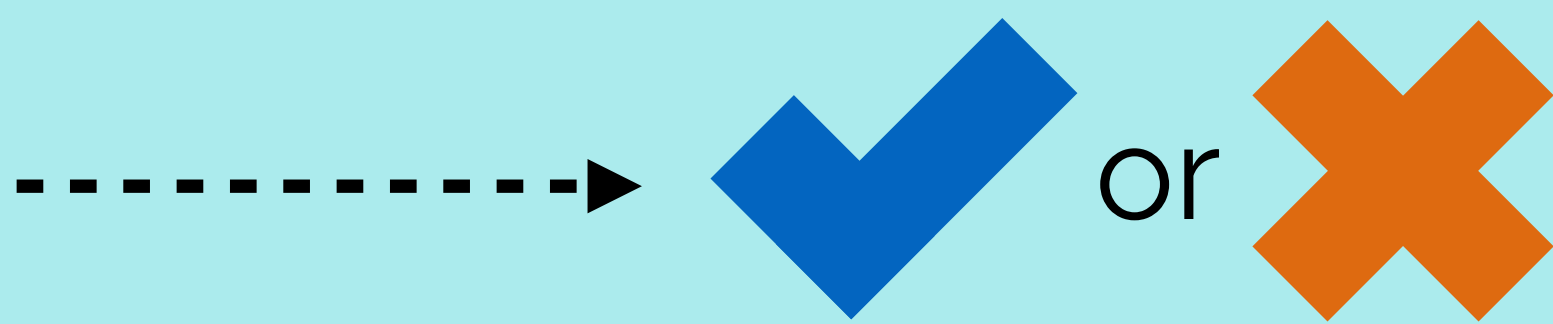
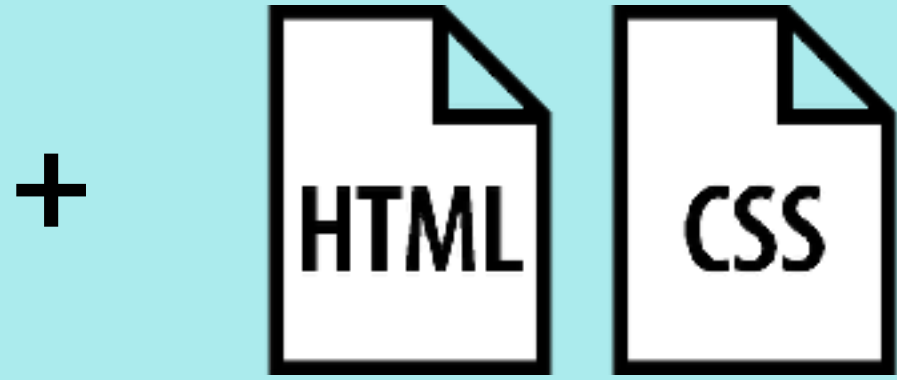


require  $layout = \text{render}_k(html, css, params)$

require  $\neg P(layout) \vee \neg \text{sufficient-}k(layout)$

# VizAssert

Guideline  
Button visible

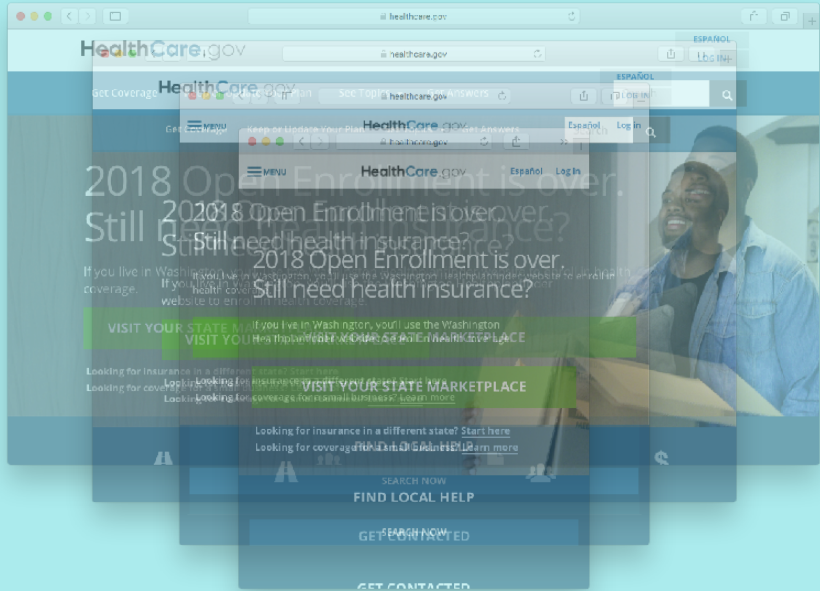


1.

$\forall b_1, b_2 \in \mathcal{B},$   
 $b_1 \in \$(.search-button) \wedge$   
 $b_2 \in \$(.toolbar) \implies$   
 $\text{within}(b_1, b_2)$

Visual Logic

2.



Renderings

3.

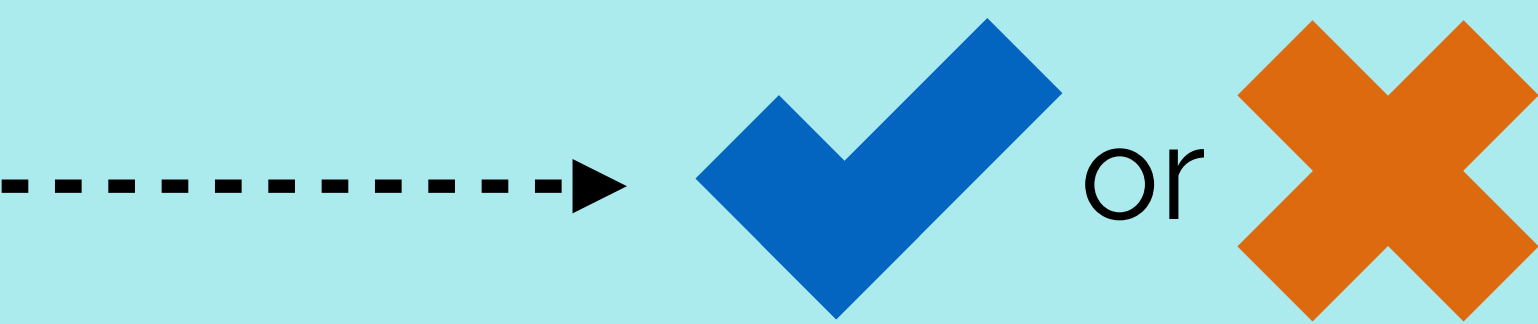
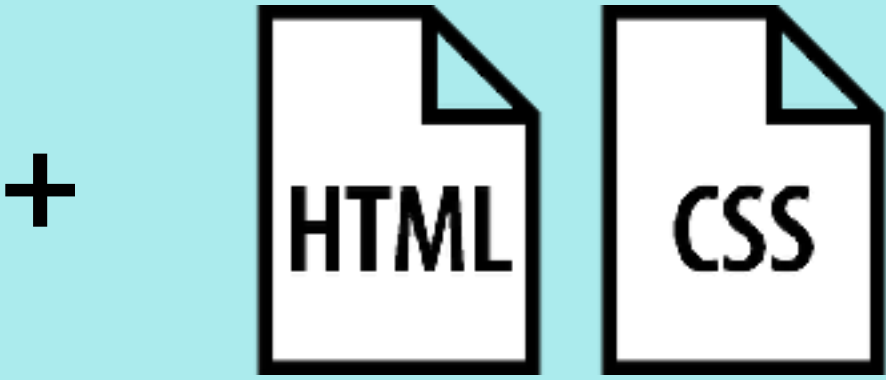
(set-logic QF\_LRA)  
(declare-type Box ...)  
(declare-const b1 Box)  
(declare-const b2 Box)  
(assert (not ...))

SMT Query

Z3

# VizAssert

Guideline  
Button visible



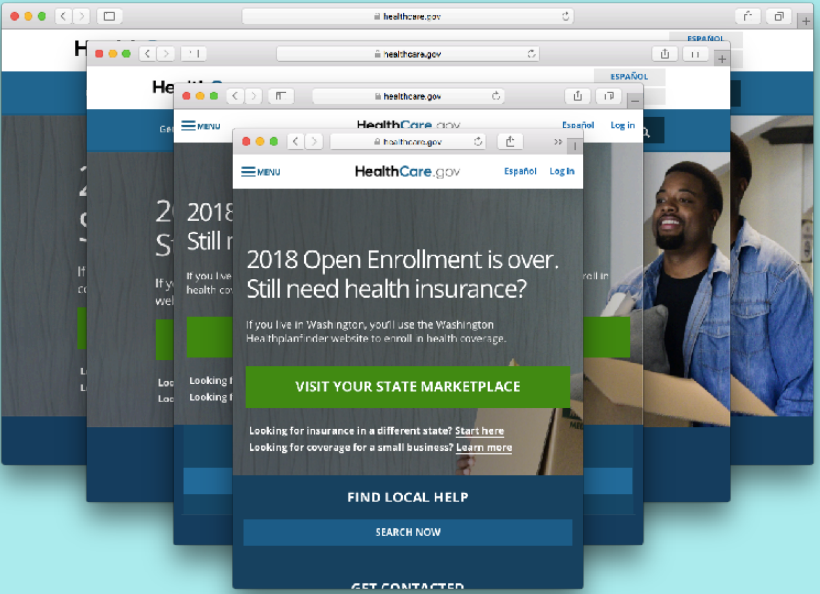
Evaluation

1.

$\forall b_1, b_2 \in \mathcal{B},$   
 $b_1 \in \$(.search-button) \wedge$   
 $b_2 \in \$(.toolbar) \implies$   
 $within(b_1, b_2)$

Visual Logic

2.



Renderings

3.

```
(set-logic QF_LRA)
(declare-type Box ...)
(declare-const b1 Box)
(declare-const b2 Box)
(assert (not ...))
```

SMT Query

## 14 Real-world Assertions

Text is at least 14px tall

Main button is big enough  *Customize to page*

Lines at most 80 characters

Text not overlapping

Contrast is sufficient

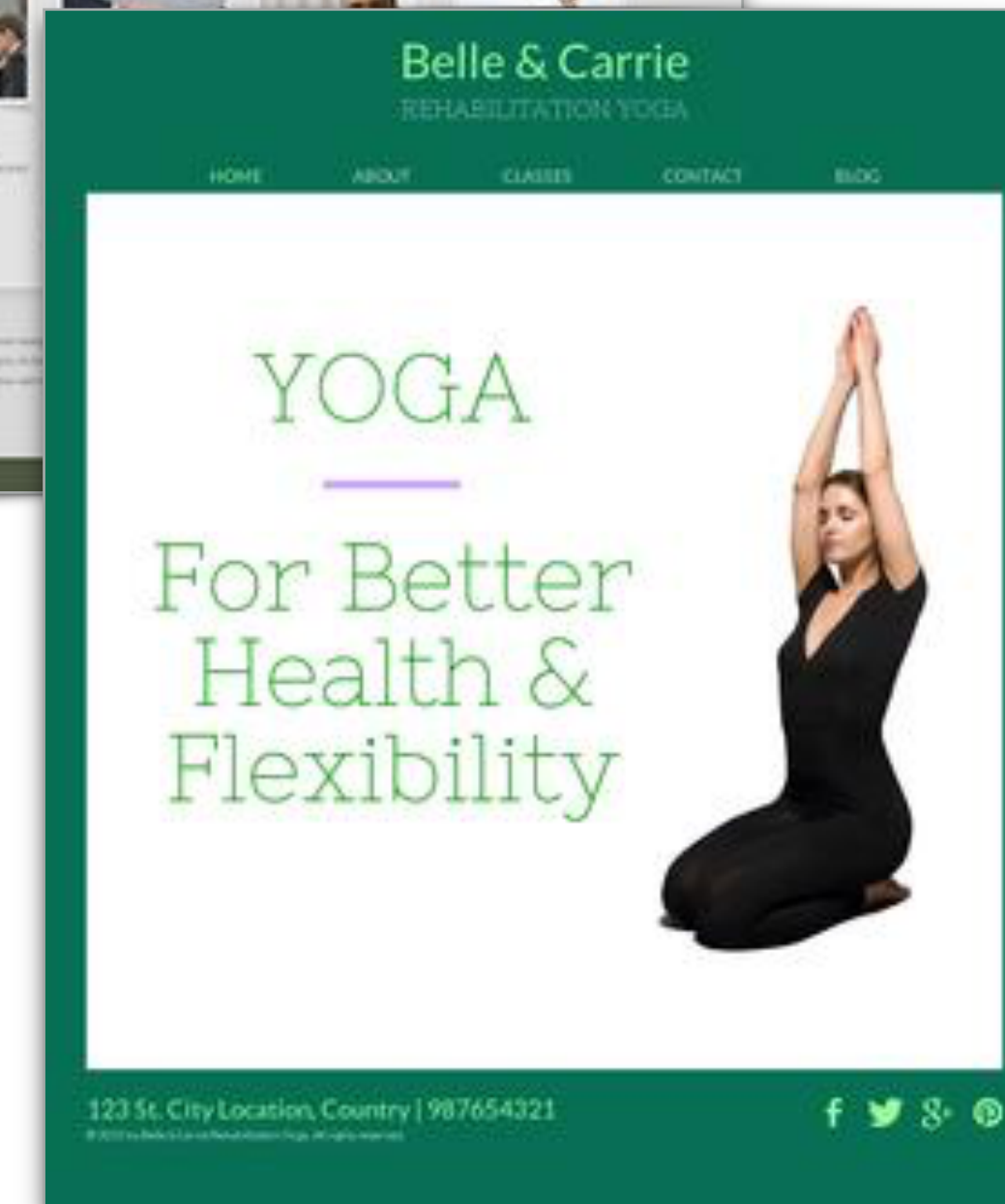
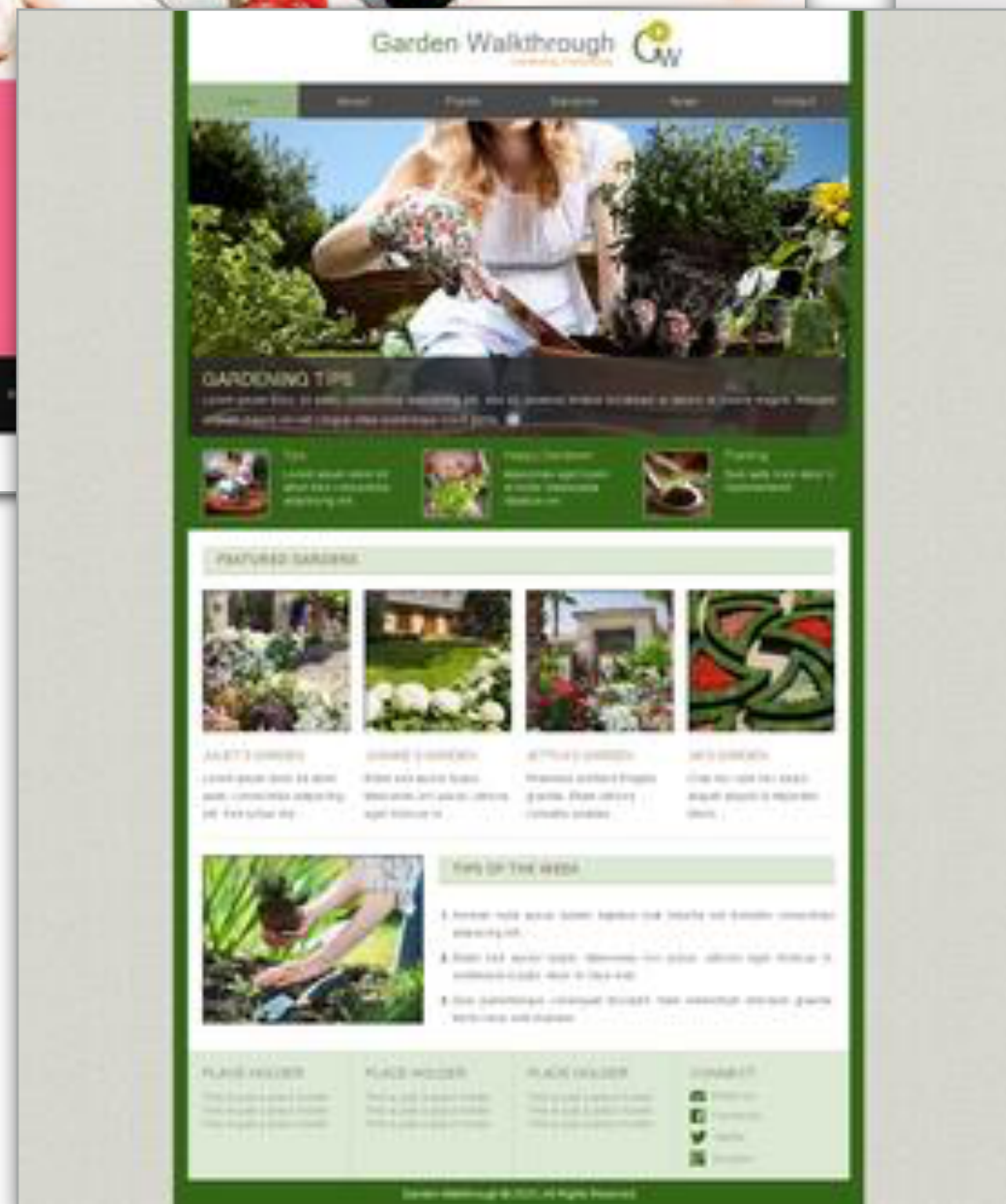
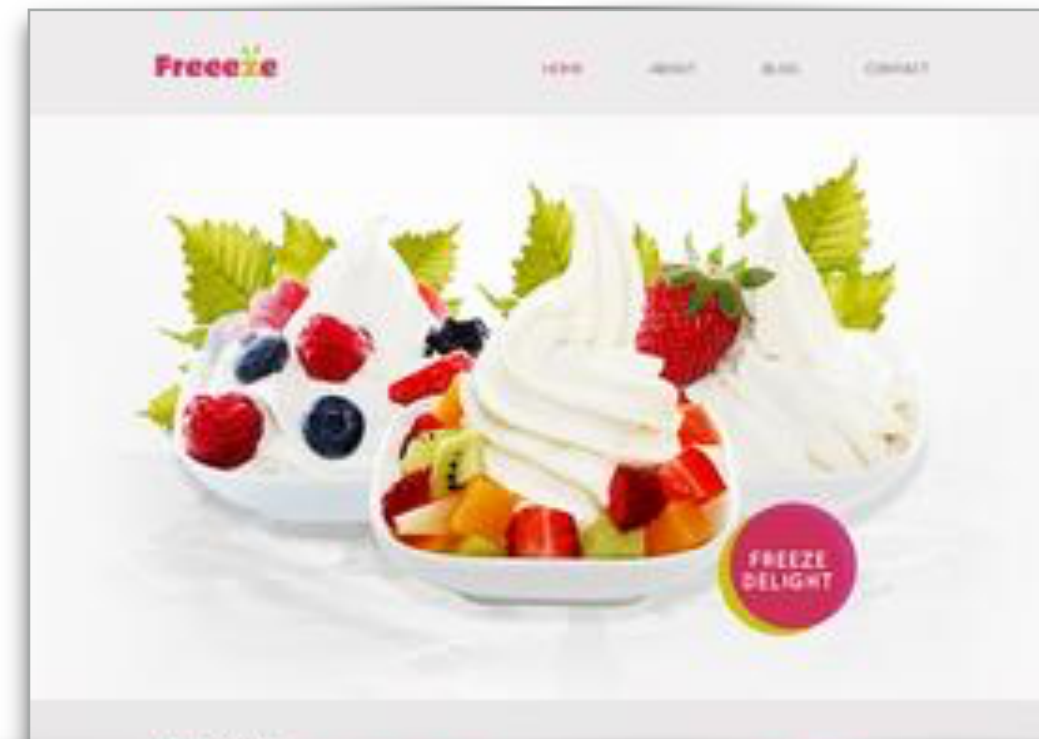
Headings form hierarchy  *Design principles*

Columns vertically aligned

...



# 62 Real-World Web Pages

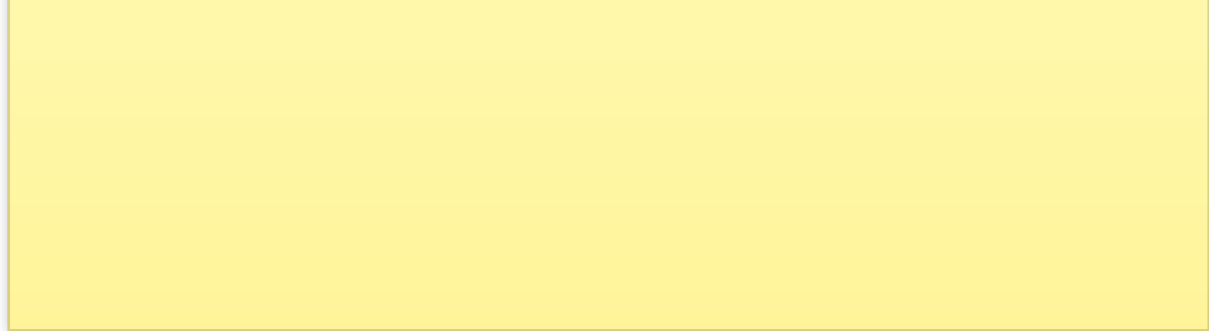


# VizAssert's Results

|                         | Assertion             | Verified   | Bug found | False Positive | Timeout   |
|-------------------------|-----------------------|------------|-----------|----------------|-----------|
| <i>No Customization</i> | Text size             | 38         | 18        | 3              | 3         |
|                         | Button position       | 59         | 1         | 1              | 1         |
|                         | Line width            | 39         | 18        | 3              | 2         |
|                         | Alt text position     | 36         |           |                |           |
|                         | No horiz. scroll      | 60         |           | 1              | 1         |
|                         | Heading sizes         | 39         | 21        |                | 2         |
|                         | Occluded text         | 53         | 2         | 5              | 2         |
|                         | Line spacing          | 59         | 3         |                |           |
| <i>Customized</i>       | High contrast         | 1          |           |                |           |
|                         | Text on bg            | 1          |           |                |           |
|                         | Hidden menu           | 1          |           |                |           |
|                         | Align columns         | 1          |           |                |           |
|                         | Visible text          |            |           | 1              |           |
|                         | Button size           | 1          |           |                |           |
|                         | <b>Total (of 476)</b> | <b>388</b> | <b>64</b> | <b>13</b>      | <b>11</b> |

*Guarantees!*

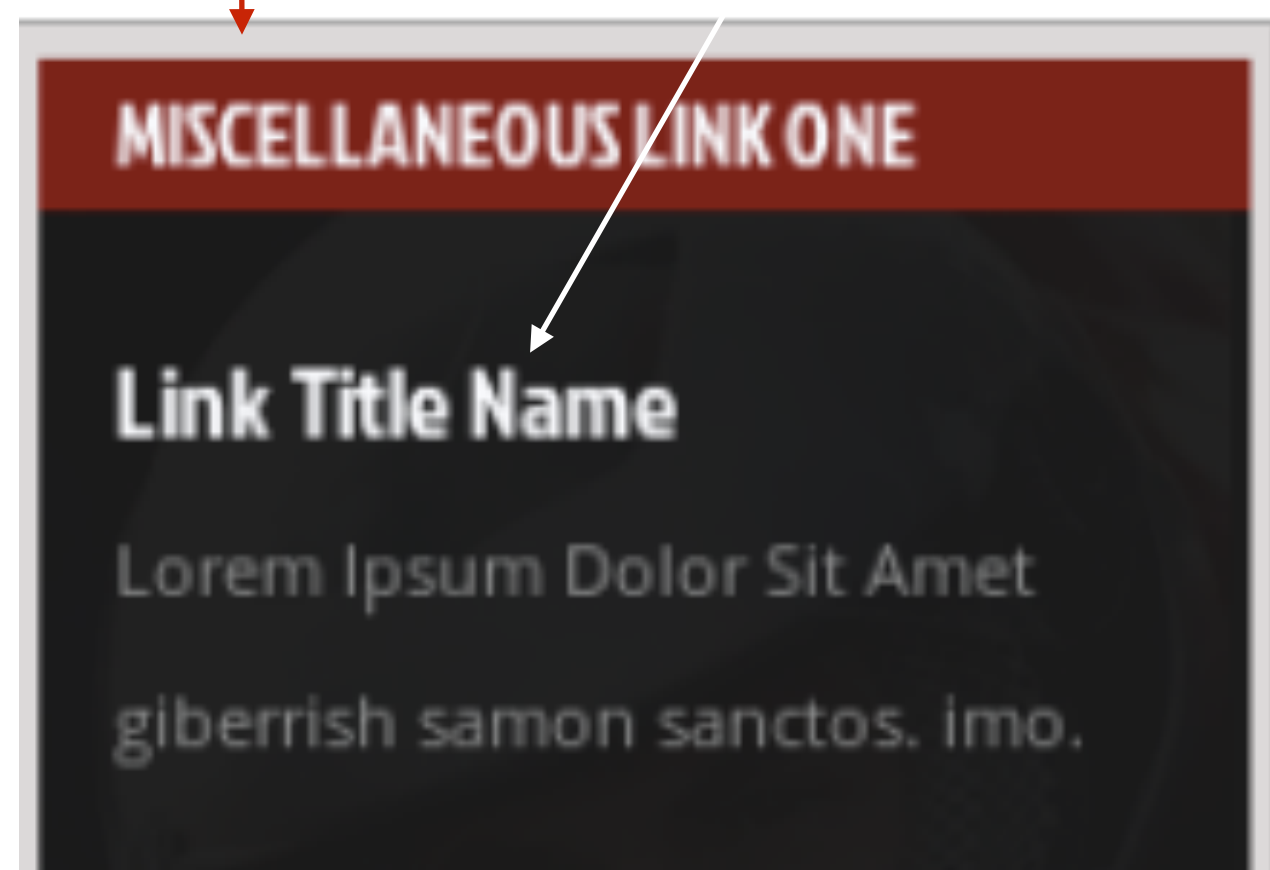
# VizAssert's Results



Timeout  
False Pos



*Small Title*  
*Big body*



True positive



False positive

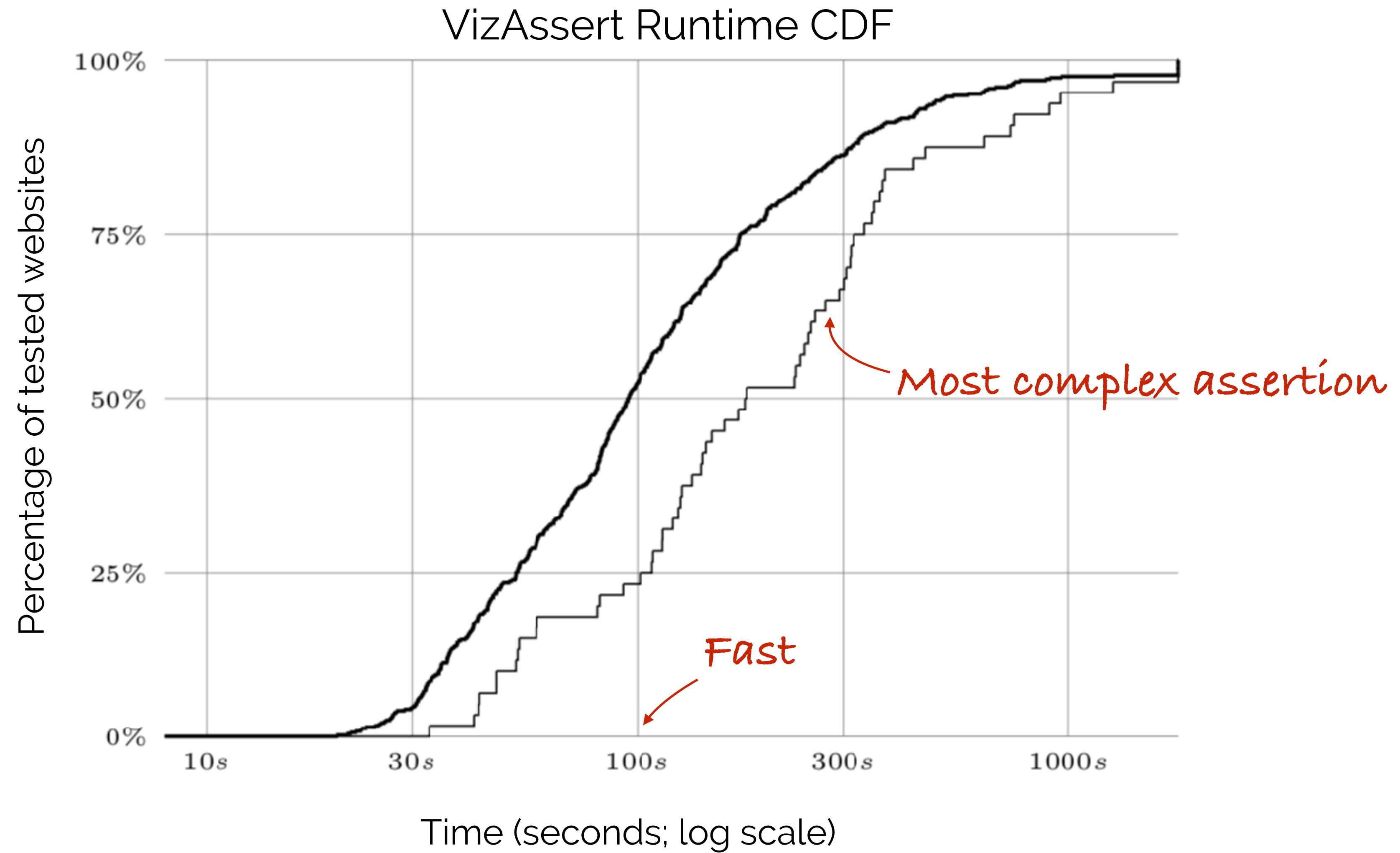
# Value of Finitization Reductions

| CSS Section       | VizAssert | OOPSLA'16 |
|-------------------|-----------|-----------|
| Margins           | 93%       | 64%       |
| Collapsing        | 96%       | 10%       |
| Floats            | 90%       | 0%        |
| Float position    | 89%       | 0%        |
| Clearance         | 87%       | 0%        |
| Line height       | 70%       | 10%       |
| Leading           | 89%       | 18%       |
| Overall (of 1009) | 91%       | 27%       |

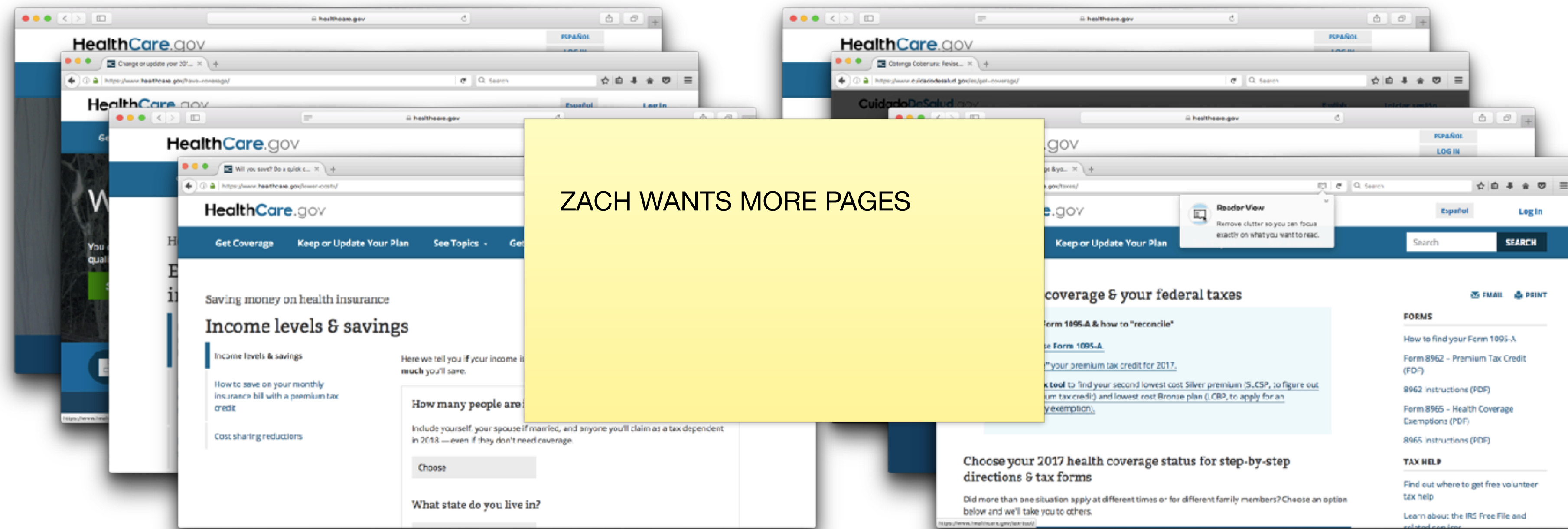
Rediscovered two known Firefox bugs

And some purposeful deviations from the standard

# Performance



# Future Work

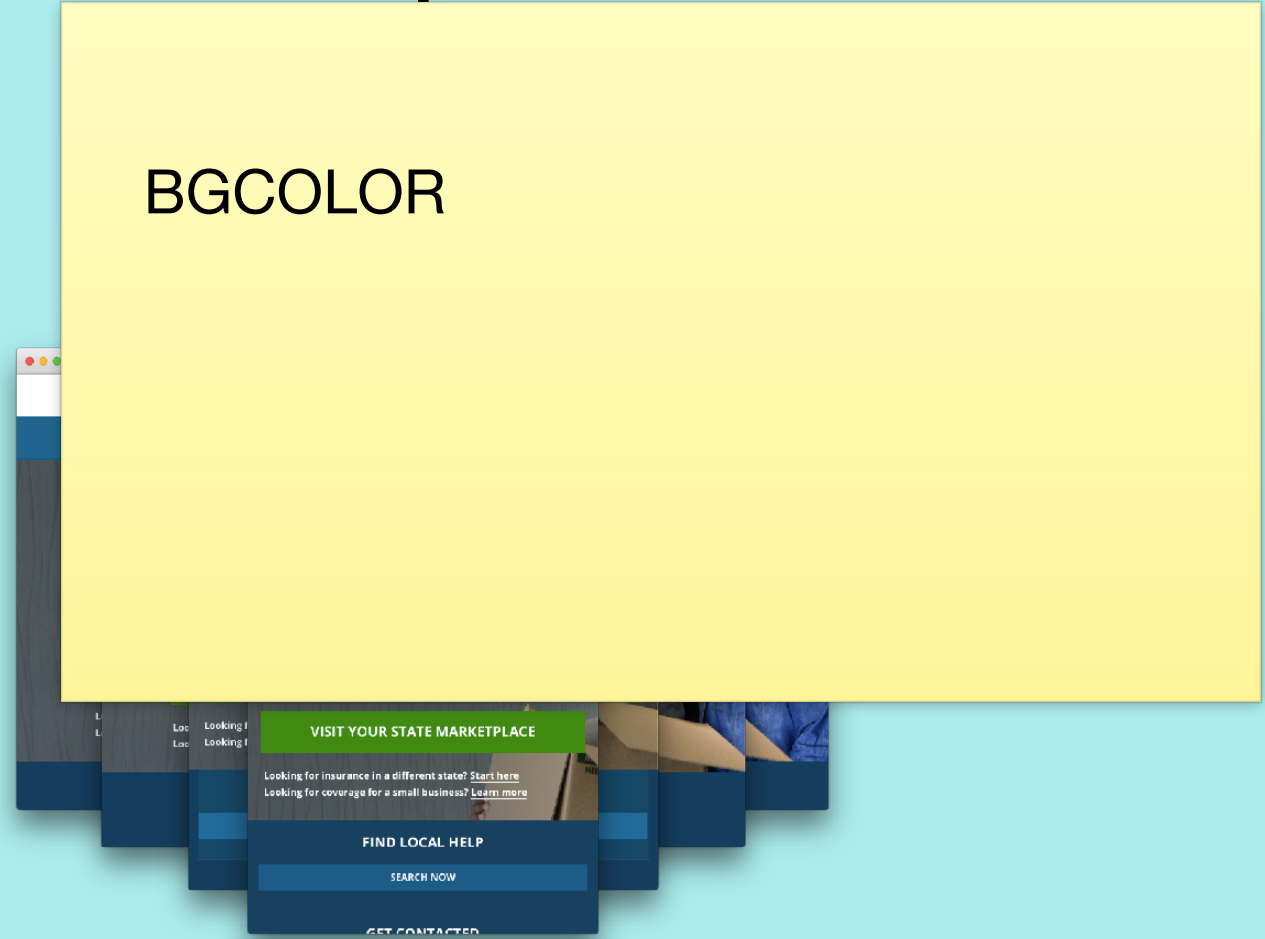
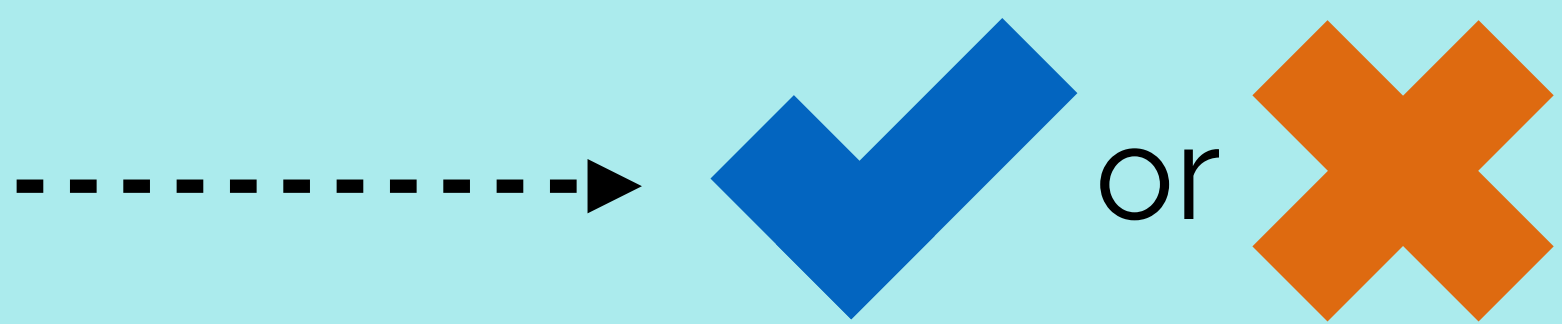
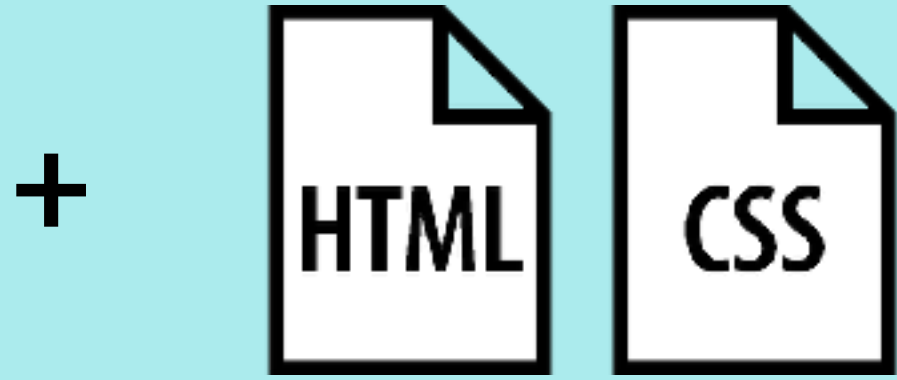


Per-component reasoning

Reuse, incrementalization, caching, parallelization

# VizAssert

Guideline  
Button visible



Evaluation

1.

$\forall b_1, b_2 \in \mathcal{B},$   
 $b_1 \in \$(.search-button) \wedge$   
 $b_2 \in \$(.toolbar) \implies$   
 $\text{within}(b_1, b_2)$

Visual Logic

Renderings

3.

```
(set-logic QF_LRA)
(declare-type Box ...)
(declare-const b1 Box)
(declare-const b2 Box)
(assert (not ...))
```

SMT Query

# Automated Verification of Web Page Layout

Visual Logic

Formalize layout properties

Novel semantics of CSS

Line height, margins, floats

Handles real web pages

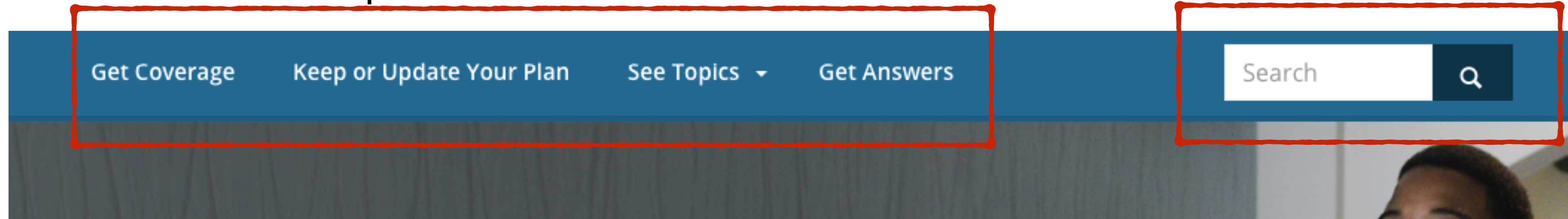
95% verified or true positive



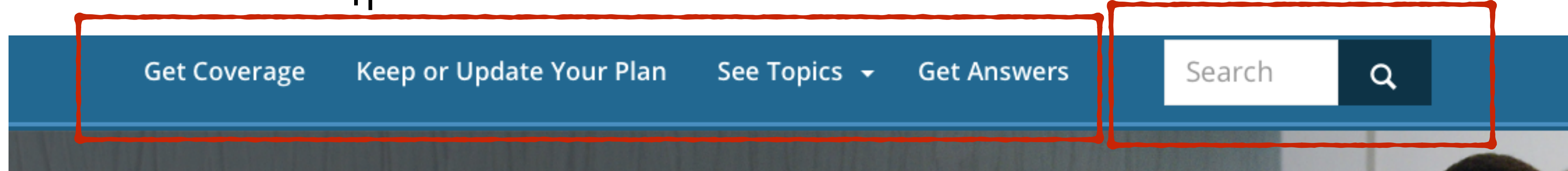
<https://cassius.uwplse.org>



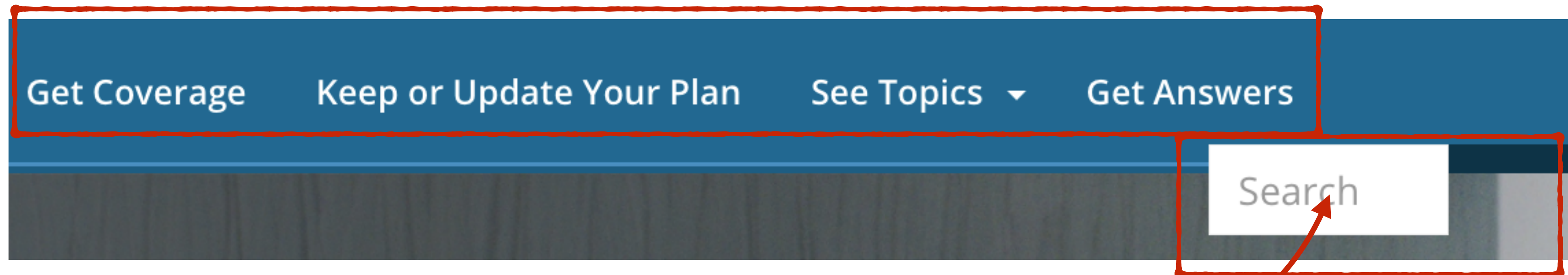
Font Size: 16px



Font Size: 24px



Font Size: 32px



*Image occludes button*