Verifying That Web Pages Have Accessible Layout

Pavel Panchekha, Adam T. Geller, Michael D. Ernst, Zachary Tatlock, Shoaib Kamil
Existing Verification Work

7% Backend
7% Layout
9% Script
17% Config
66% This Work: Automated Verification for Web Page Layout

3.6M lines
Usability, accessibility, mobile-friendliness
Complex Behavior + Correctness Properties

VizAssert

Automatic Verification for Web Page Layout
VizAssert

Guideline Button visible + HTML CSS

Over configurable range of:
Browser widths
Browser heights
Font sizes
Scrollbar widths
....

Sound Guarantee
Plus Concrete Rendering
Guideline Button visible

1. Visual Logic

∀b₁, b₂ ∈ B,
b₁ ∈ $(.search-button) ∧
b₂ ∈ $(.toolbar) ⇒
within(b₁, b₂)

2. Renderings

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

3. SMT Query

VizAssert

+ HTML CSS

or

Z₃

Button visible
VizAssert

Guideline
Button visible

∀b₁, b₂ ∈ B,
b₁ ∈ $(.search-button) ∧
b₂ ∈ $(.toolbar) →
within(b₁, b₂)

Visual Logic

+ HTML CSS

Renderings

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

SMT Query

Z3
VizAssert

1. Visual Logic
   \[ \forall b_1, b_2 \in B, \]
   \[ b_1 \in \$(\cdot.search-button) \land \]
   \[ b_2 \in \$(\cdot.toolbar) \Rightarrow \]
   \[ \text{within}(b_1, b_2) \]

2. Renderings

3. SMT Query
   (set-logic QF_LRA)
   (declare-type Box ...)
   (declare-const b1 Box)
   (declare-const b2 Box)
   (assert (not ...))
Guideline

Button visible

Visual Logic

\[ \forall b_1, b_2 \in \mathcal{B}, \]
\[ b_1 \in \$(\text{.search-button}) \land \]
\[ b_2 \in \$(\text{.toolbar}) \implies \]
\[ \text{within}(b_1, b_2) \]
Universal quantifier

\[ \forall b_1, b_2, \ldots \]

**HTML**
- root
- b.parent
- b.previous
- b.first-child
- \( b \in \$(selector) \)

**CSS**
- b.left
- b.top
- b.previous
- \( x + 2y - z \)
- \( b_1.top < b_2.top \)
- b.bg.r

**Visual Logic**
VizAssert

1. Guideline
   Button visible
   \[ \forall b_1, b_2 \in B, \]
   \[ b_1 \in \$(\text{.search-button}) \land \]
   \[ b_2 \in \$(\text{.toolbar}) \implies \]
   \[ \text{within}(b_1, b_2) \]
   Visual Logic

2. Renderings
   HTML CSS

3. SMT Query
   (set-logic QF_LRA)
   (declare-type Box ...)
   (declare-const b1 Box)
   (declare-const b2 Box)
   (assert (not ...))
   Z3

or

✓ or ✗
1. Visual Logic
∀b₁, b₂ ∈ B, b₁ ∈ $(.search-button) ∧ b₂ ∈ $(.toolbar) → within(b₁, b₂)

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

3. SMT Query

Guideline
Button visible

VizAssert

or

HTML
CSS

Z₃
Symbolic web browser in SMT

102 standards
100k+ words
487 properties
Formalized CSS Features

- Styles
- Cascade
- Selectors

- Box Types
- Layout Mode
- Vertical

- Flow Width
- Horizontal
- Height

- Width
- Shrink-to-fit
- Line Height

- Margin Collapse
- Floating Layout
- Clearance
Finitization Reductions

\[ \forall x \exists y \exists z = y \quad \text{Quantifier alternation} \]

\[ \mu x. P(x) \quad \text{Fixpoints, reachability} \]

\[ \text{sort}(L) \quad \text{Data structures} \]

\[ \exists z = y \quad \text{Line Height} \]

\[ \text{Margin Collapsing} \]

\[ \text{Floating Layout} \]

Finite number of registers
Line Height

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

Reduced to 2-register encoding

Incremental maximum/minimum computations
Margin collapsing

“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

“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.”
Float Semantics

Where floats are allowed
Float Semantics
Float Semantics

Text Line will wrap around floats.
Float Semantics

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

In practice, \(k \leq 7\); search for sufficient value
Finitization Reductions

\[ \forall E \exists A \forall A \exists E \] Quantifier alternation
  
  - Line Height \quad \rightarrow 2 \text{ registers}

\[ \mu x. P(x) \] Fixpoints, reachability
  
  - Margin Collapsing \quad \rightarrow 5 \text{ registers}

\[ \text{sort}(L) \] Data structures
  
  - Floating Layout \quad \rightarrow 5k+2 \text{ registers}
VizAssert

1. Visual Logic
   \[ \forall b_1, b_2 \in B, b_1 \in \{.search-button\} \land b_2 \in \{.toolbar\} \implies \text{within}(b_1, b_2) \]

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

3. SMT Query

Guideline
Button visible

+ HTML
CSS

\[
\begin{align*}
3. & \text{SMT Query} \\
2. & \text{Renderings} \\
1. & \text{Visual Logic}
\end{align*}
\]
VizAssert

Guideline

Button visible

Guideline

Button visible

∀b₁, b₂ ∈ B, b₁ ∈ $(.search-button) ∧ b₂ ∈ $(.toolbar) → within(b₁, b₂)

Visual Logic

1.

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

SMT Query

3.

Renderings

2.

HTML

CSS

+
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 ¬ P(layout)
SMT Query

val html = "<html><body>…</body></html>"
val css = "#main { … }; p li { … }; …"

require layout = render (html, css, params)
require ¬ P(layout) ∨ ¬ sufficient-\(k\)(layout)
VizAssert

1. Visual Logic
   \[ \forall b_1, b_2 \in B, \]
   \[ b_1 \in \$(\text{.search-button}) \land \\
   b_2 \in \$(\text{.toolbar}) \implies \\
   \text{within}(b_1, b_2) \]

   Guideline
   Button visible

2. Renderings

3. SMT Query
   (set-logic QF_LRA)
   (declare-type Box ...)
   (declare-const b1 Box)
   (declare-const b2 Box)
   (assert (not ...))

or
VizAssert

Guideline
Button visible

Visual Logic
\[ \forall b_1, b_2 \in B, b_1 \in $\cdot$search-button$\cdot$ \land b_2 \in $\cdot$toolbar$\cdot$ \implies \text{within}(b_1, b_2) \]

1.

Renderings

2.

HTML
CSS

or

3.

Evaluation

(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
Lines at most 80 characters
Text not overlapping
Contrast is sufficient
Headings form hierarchy
Columns vertically aligned

...
62 Real-World Web Pages
### VizAssert's Results

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Verified</th>
<th>Bug found</th>
<th>False Positive</th>
<th>Timeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text size</td>
<td>38</td>
<td>18</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Button position</td>
<td>59</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Line width</td>
<td>39</td>
<td>18</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>Alt text position</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>No horiz. scroll</td>
<td>60</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Heading sizes</td>
<td>39</td>
<td>21</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>Occluded text</td>
<td>53</td>
<td>2</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>Line spacing</td>
<td>59</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>High contrast</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Text on bg</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Hidden menu</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Align columns</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Visible text</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Button size</td>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Total (of 476)</td>
<td>388</td>
<td>64</td>
<td>13</td>
<td>11</td>
</tr>
</tbody>
</table>
VizAssert's Results

Verified

Small Title

Big body

True positive

False positive

Timeout

False Pos

True Pos
## Value of Finitization Reductions

<table>
<thead>
<tr>
<th>CSS Section</th>
<th>VizAssert</th>
<th>OOPSLA’16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Margins</td>
<td>93%</td>
<td>64%</td>
</tr>
<tr>
<td>Collapsing</td>
<td>96%</td>
<td>10%</td>
</tr>
<tr>
<td>Floats</td>
<td>90%</td>
<td>0%</td>
</tr>
<tr>
<td>Float position</td>
<td>89%</td>
<td>0%</td>
</tr>
<tr>
<td>Clearance</td>
<td>87%</td>
<td>0%</td>
</tr>
<tr>
<td>Line height</td>
<td>70%</td>
<td>10%</td>
</tr>
<tr>
<td>Leading</td>
<td>89%</td>
<td>18%</td>
</tr>
<tr>
<td>Overall (of 1009)</td>
<td>91%</td>
<td>27%</td>
</tr>
</tbody>
</table>

Rediscovered two known Firefox bugs

And some purposeful deviations from the standard
Performance

VizAssert Runtime CDF

Most complex assertion

Fast
Future Work

Per-component reasoning
Reuse, incrementalization, caching, parallelization
1. Visual Logic

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

2. SMT Query

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

3. Evaluation

Guideline

Button visible

HTML

CSS

BGCOLOR

VizAssert
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