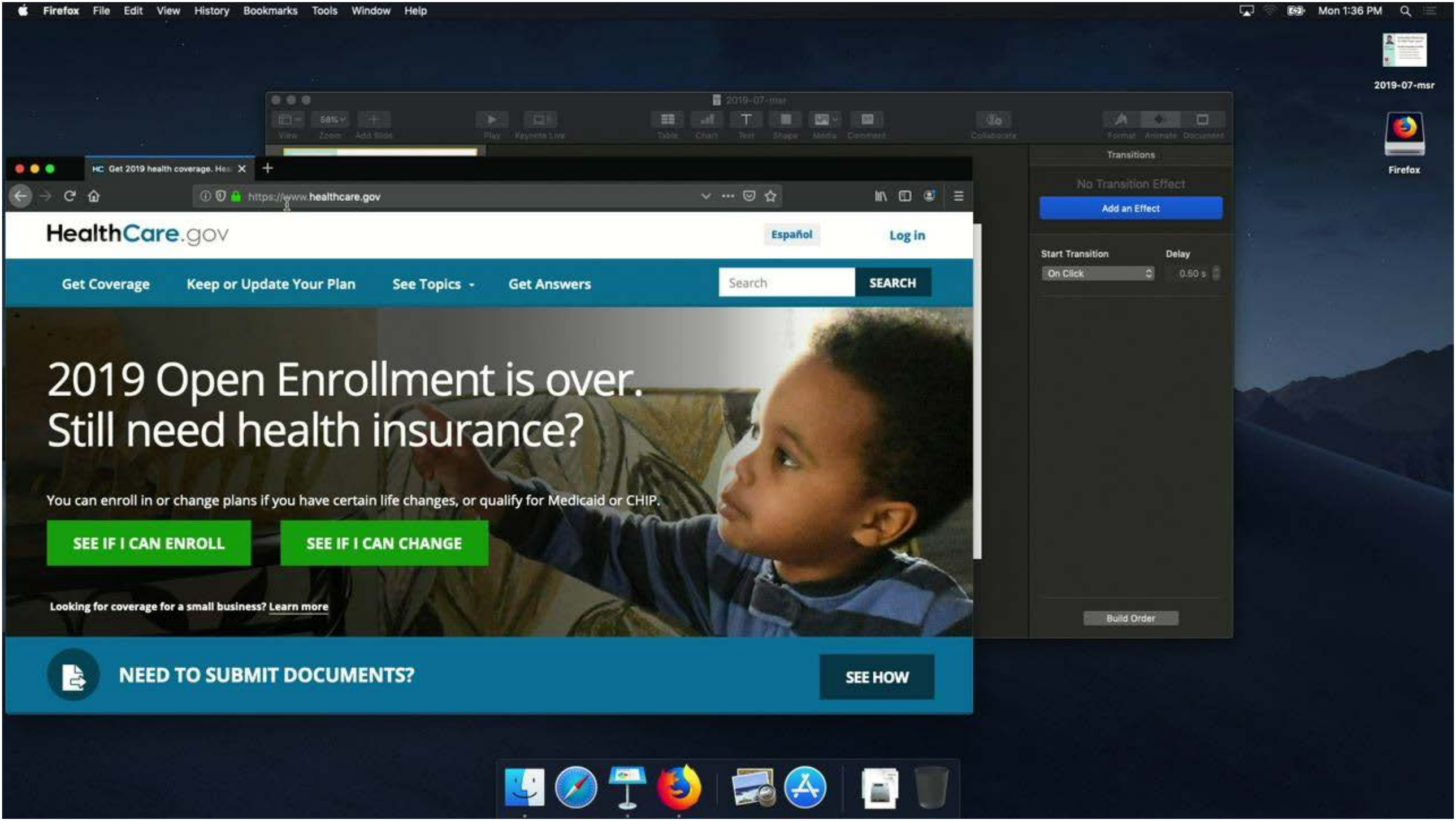
Pavel
Panchekha

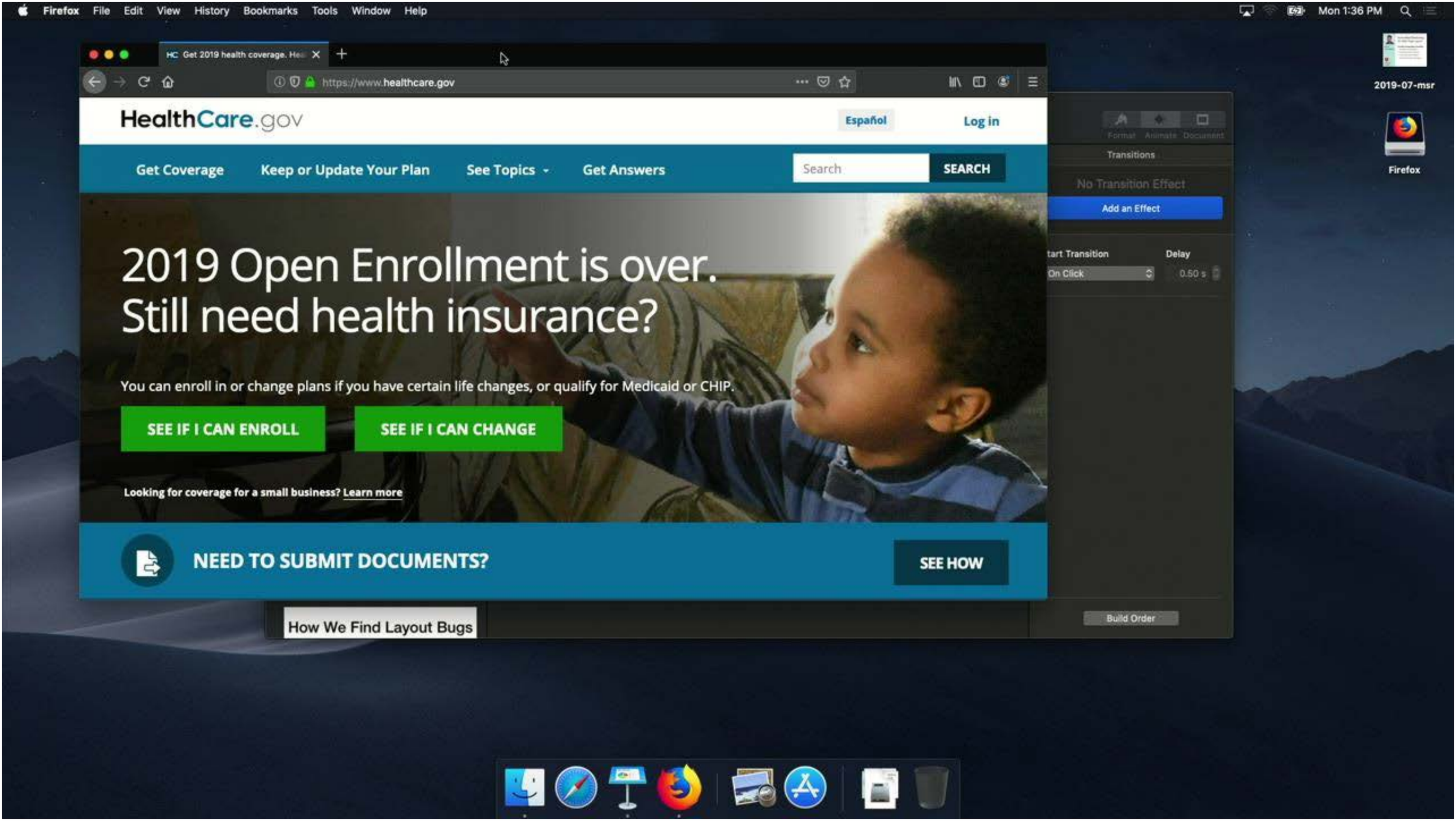
University
of Utah

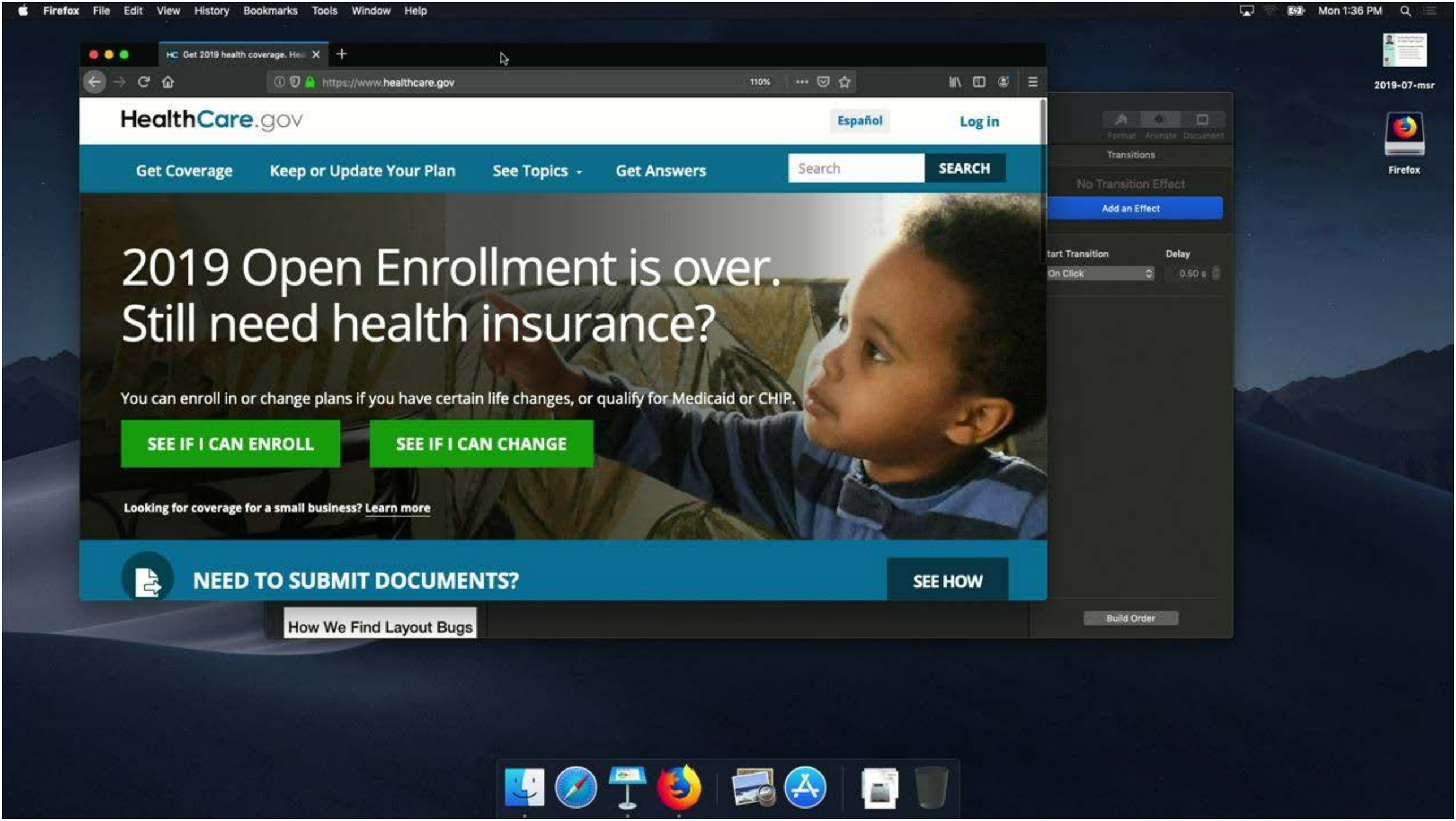
Automated Reasoning for Web Page Layout

Formally proving pages accessible:

- Specifying visual properties
- Formalizing browser rendering
- Overcoming solver limitations
- Finding modularity in web pages


Firefox File Edit View History Bookmarks Tools Window Help
Mon 1:36 PM
2019-07-msr
Firefox
2019-07-msr
Transitions
No Transition Effect
Add an Effect
Start Transition
Delay
On Click
0.50 s
Build Order
HC Get 2019 health coverage. Hea
https://www.healthcare.gov
HealthCare.gov
Español
Log in
Get Coverage
Keep or Update Your Plan
See Topics
Get Answers
Search
SEARCH
2019 Open Enrollment is over.
Still need health insurance?
You can enroll in or change plans if you have certain life changes, or qualify for Medicaid or CHIP.
SEE IF I CAN ENROLL
SEE IF I CAN CHANGE
Looking for coverage for a small business? Learn more
NEED TO SUBMIT DOCUMENTS?
SEE HOW


Firefox File Edit View History Bookmarks Tools Window Help
Mon 1:36 PM
2019-07-msr
Firefox
HC Get 2019 health coverage. He
https://www.healthcare.gov
HealthCare.gov
Español
Log in
Get Coverage
Keep or Update Your Plan
See Topics
Get Answers
Search
SEARCH
2019 Open Enrollment is over.
Still need health insurance?
You can enroll in or change plans if you have certain life changes, or qualify for Medicaid or CHIP.
SEE IF I CAN ENROLL
SEE IF I CAN CHANGE
Looking for coverage for a small business? Learn more
NEED TO SUBMIT DOCUMENTS?
SEE HOW
How We Find Layout Bugs
Format
Animate
Document
Transitions
No Transition Effect
Add an Effect
tart Transition
Delay
On Click
0.50 s
Build Order


Firefox File Edit View History Bookmarks Tools Window Help
Mon 1:36 PM
2019-07-msr
Firefox
HC Get 2019 health coverage. He
https://www.healthcare.gov
110%
HealthCare.gov
Español
Log in
Get Coverage
Keep or Update Your Plan
See Topics
Get Answers
Search
SEARCH
2019 Open Enrollment is over.
Still need health insurance?
You can enroll in or change plans if you have certain life changes, or qualify for Medicaid or CHIP.
SEE IF I CAN ENROLL
SEE IF I CAN CHANGE
Looking for coverage for a small business? Learn more
NEED TO SUBMIT DOCUMENTS?
SEE HOW
How We Find Layout Bugs
Transitions
No Transition Effect
Add an Effect
Delay
On Click
0.50 s
Build Order

Layout Bugs are Endemic

HealthCare.gov
Missing button

Bank of America
Hidden text

FAFSA
Overlapping text

Walgreens
Hidden tabs

Layout Bugs are Endemic

HealthCare.gov
Missing button

Bank of America
Hidden text

FAFSA
Overlapping text

Walgreens
Hidden tabs

Layout Bugs are Endemic

HealthCare.gov
Missing button

Bank of America
Hidden text

FAFSA
Overlapping text

Walgreens
Hidden tabs

12% of Americans have disabilities

Layout Bugs are Endemic

HealthCare.gov
Missing button

Bank of America
Hidden text

FAFSA
Overlapping text

Walgreens
Hidden tabs

12% of Americans have disabilities

Accommodation legally required (under ADA)

How We Find Layout Bugs

Industry standard:

Manual Testing

Manual review
of renderings

Manual selection
of configurations

How We Find Layout Bugs

Industry standard:
Manual Testing

Manual review
of renderings

Manual selection
of configurations

State of the Art:
Automated Tests

Auto-comparison
of renderings

Random test array
of configurations

How We Find Layout Bugs

Industry standard:
Manual Testing

Manual review
of renderings

Manual selection
of configurations

State of the Art:
Automated Tests

Auto-comparison
of renderings

Random test array
of configurations

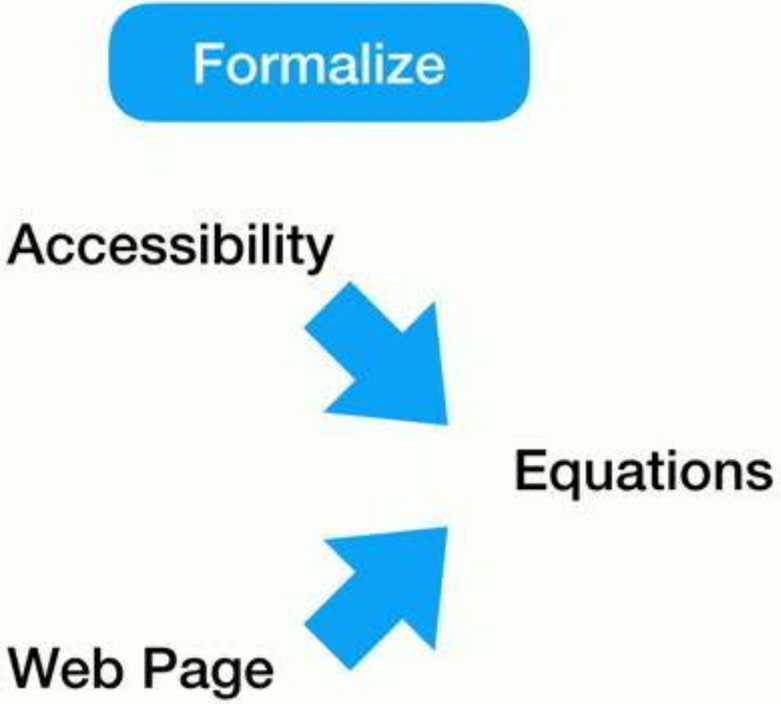
My Work:
Verification

Formal specification
for valid renderings

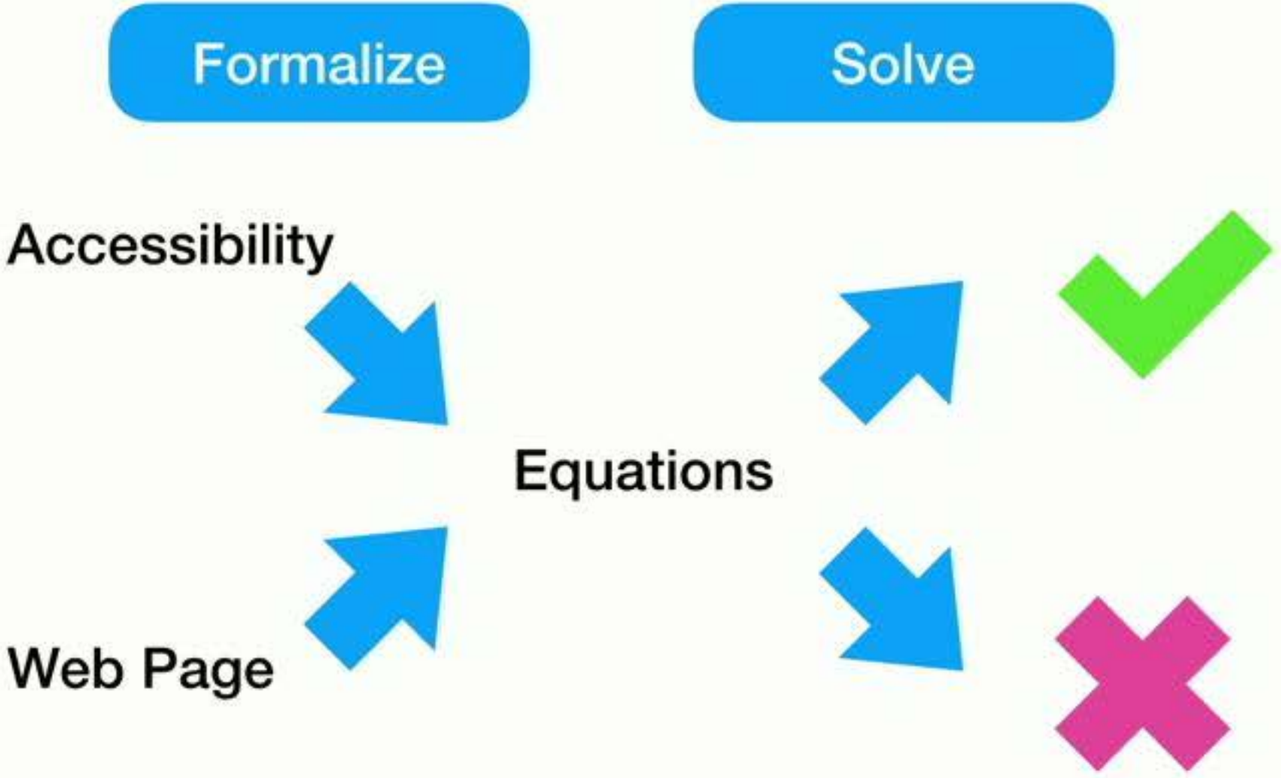
Automated proof
for all configurations

How VizAssert Works

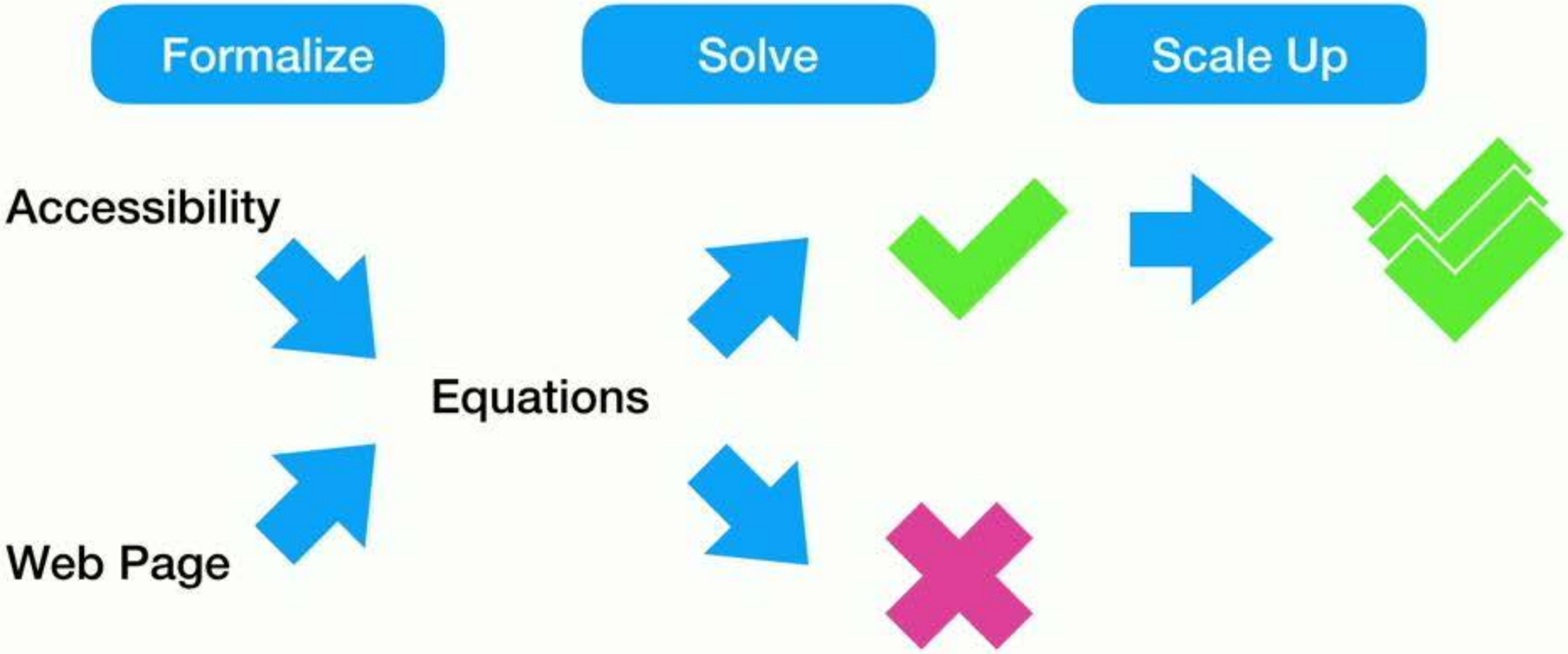
How VizAssert Works



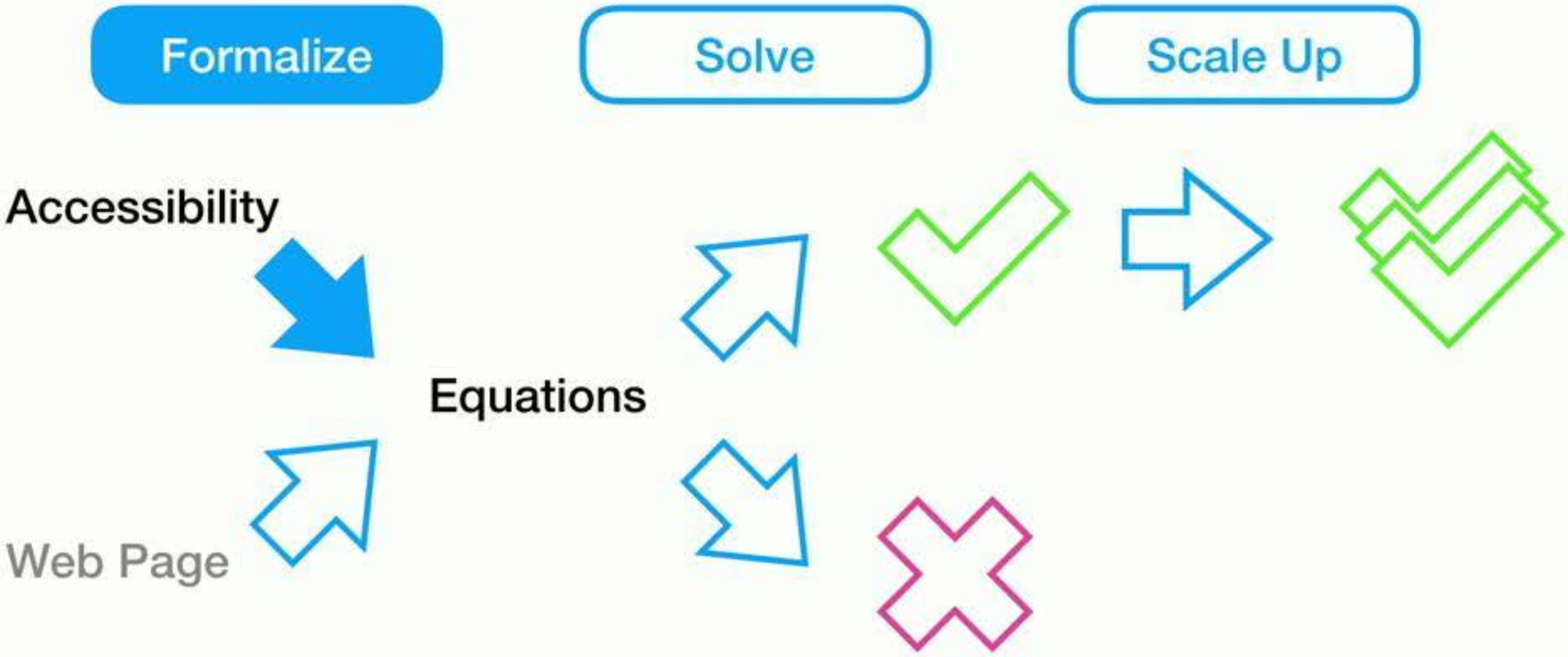
How VizAssert Works



How VizAssert Works



How VizAssert Works



Web Pages

HTML + CSS + JS

Web Pages

HTML + CSS

- Attractive
- Easy to Use
- Mobile Friendly
- Accessible
- Simple

Web Pages

HTML + CSS

- Attractive
- Easy to Use
- Mobile Friendly
- **Accessible**
- Simple

Accessibility Guidelines

Developed by
accessibility researchers

Accessibility Guidelines

Developed by
accessibility researchers

All handled by VizAssert

Size & position guidelines

- Text is at least 14px tall
- Lines are at most 80 chars
- Text doesn't overlap

Screen-reader assistance

- Screen-reader text is offscreen
- Header hierarchy matches sizes

Functionality guidelines

- Text zoom up to 200%
- Scrolling in only one dimension

Accessibility Guidelines

Developed by
accessibility researchers

Select page elements

All handled by VizAssert

Size & position guidelines

- Text is at least 14px tall
- Lines are at most 80 chars
- Text doesn't overlap

Screen-reader assistance

- Screen-reader text is offscreen
- Header hierarchy matches sizes

Functionality guidelines

- Text zoom up to 200%
- Scrolling in only one dimension

Accessibility Guidelines

Developed by accessibility researchers

Select page elements

Constrain geometry

All handled by VizAssert

Size & position guidelines

- Text is at least 14px tall
- Lines are at most 80 chars
- Text doesn't overlap

Screen-reader assistance

- Screen-reader text is offscreen
- Header hierarchy matches sizes

Functionality guidelines

- Text zoom up to 200%
- Scrolling in only one dimension

Accessibility Guidelines

Guideline

Text zoom up to 200%
with same functionality

HealthCare.gov
Missing button

Accessibility Guidelines

Guideline

Text zoom up to 200%
with same functionality

Behavior

The search bar
and search button
must be inside
the toolbar
(for text zoom ≤ 2)

HealthCare.gov
Missing button

Accessibility Guidelines

Guideline

Text zoom up to 200%
with same functionality

Behavior

The search bar
and search button
must be inside
the toolbar
(for text zoom ≤ 2)

HealthCare.gov
Missing button

Formal Specifications

Behavior

The search bar
and search button
must be inside
the toolbar
(for text zoom ≤ 2)

HealthCare.gov
Missing button

Formal Specifications

Behavior

CSS Selector

div[role=banner] [type]

must be inside

#header + div + div

(for text zoom ≤ 2)

CSS Selector

HealthCare.gov
Missing button

Formal Specifications

Behavior

CSS Selector

div[role=banner] [type]

geometry.within

Geometric primitive

#header + div + div

(for text zoom ≤ 2)

CSS Selector

HealthCare.gov
Missing button

Formal Specifications

Behavior

The search bar
and search button
must be inside
the toolbar
(for text zoom ≤ 2)

HealthCare.gov
Missing button

Formalize →

Visual Logic

all elements matching
div[role=banner] [type]
are geometry.within
the #header + div + div
when text.zoom ≤ 200%

Formal Specifications

Behavior

The search bar
and search button
must be inside
the toolbar
(for text zoom ≤ 2)

HealthCare.gov
Missing button

Formalize →

Visual Logic

all elements matching
div[role=banner] [type]
are geometry.within
the #header + div + div
when text.zoom ≤ 200%,
window.width ≥ 800

Only on desktop

Specification to Equations

Visual Logic

all elements matching

div[role=banner] [type]

are geometry.within

the #header + div + div

when text.zoom ≤ 200%,

window.width ≥ 800

Specification to Equations

Visual Logic

all elements matching

div[role=banner] [type]

are geometry.within

the #header + div + div

when text.zoom ≤ 200%,
window.width ≥ 800

Equations over Renderings

within(B, A) =

B.left ≥ A.left &&

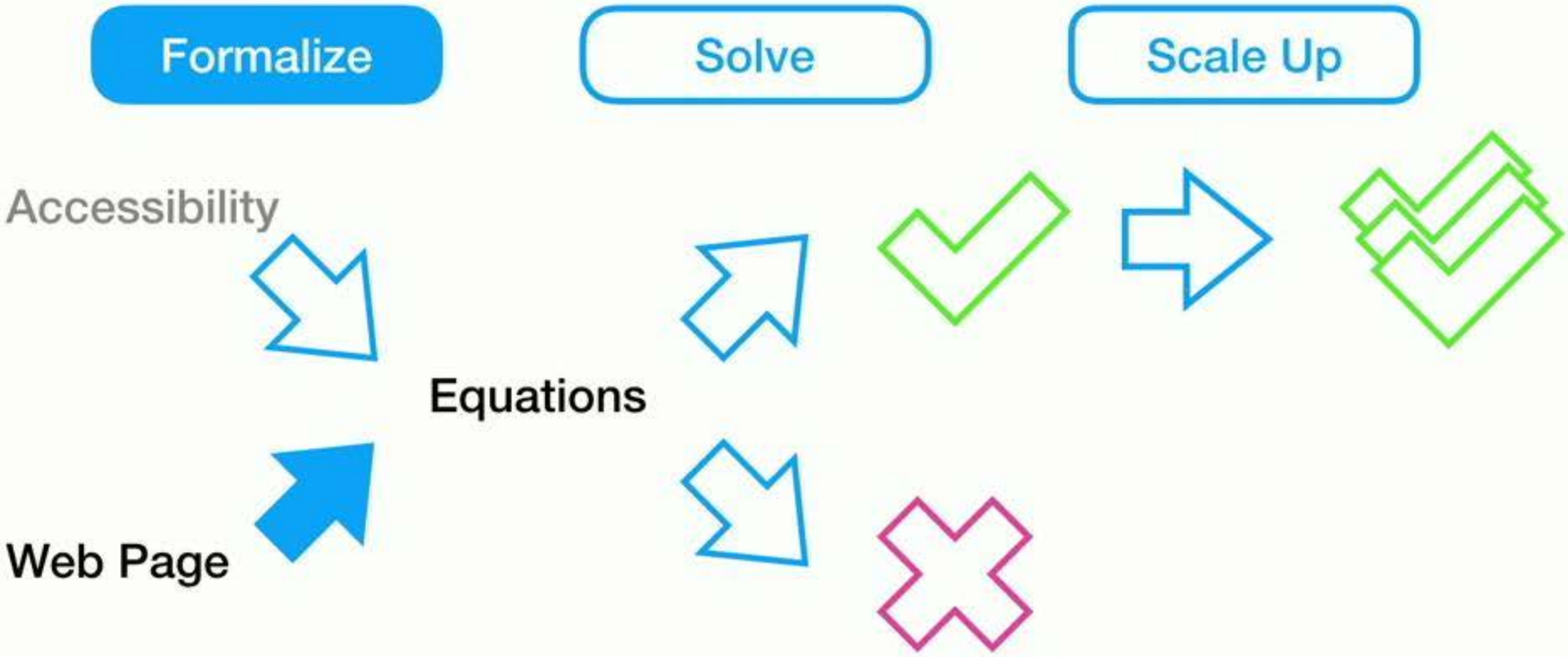
B.top ≥ A.top &&

B.left + B.width ≤ A.left + A.width &&

B.top + B.height ≤ A.top + A.height

Sizes & positions of elements

How VizAssert Works



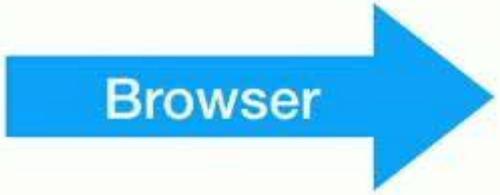
Web Browser

Web Page

HTML + CSS

Configuration

Size: 1920×1280
Text zoom: 1.5×



Rendering

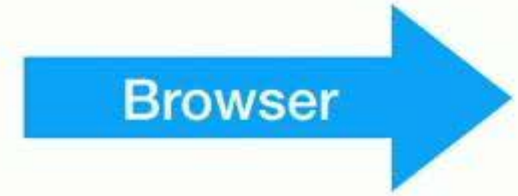
Web Browser

Web Page

HTML + CSS

Configuration

Size: 1920×1280
Text zoom: 1.5×



Rendering

Web Browser

Modern Web Browsers:

- Millions of lines of code
- Decades of development
- Dozens of developers

Symbolic Web Browser

Web Page

HTML + CSS

Configuration

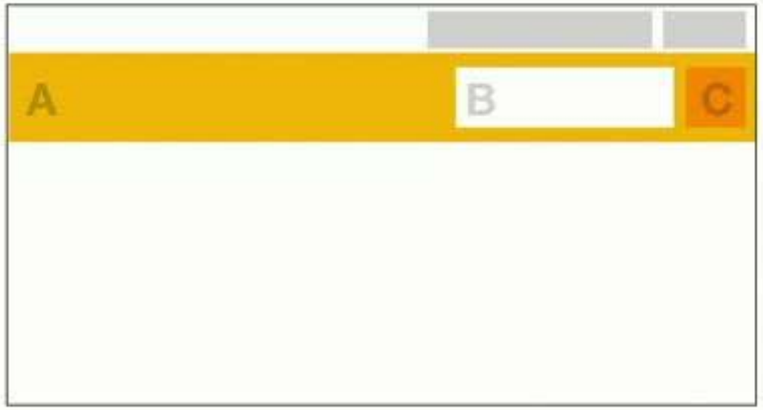
Size: $W \times H$

Text zoom: $Z\times$

Symbolic

Browser

Rendering



Symbolic Web Browser

Web Page

HTML + CSS

Configuration

Size: $W \times H$

Text zoom: $Z\times$

Symbolic

Symbolic

Browser

Rendering

Symbolic Web Browser

Web Page

HTML + CSS

Configuration

Size: $W \times H$

Text zoom: $Z\times$

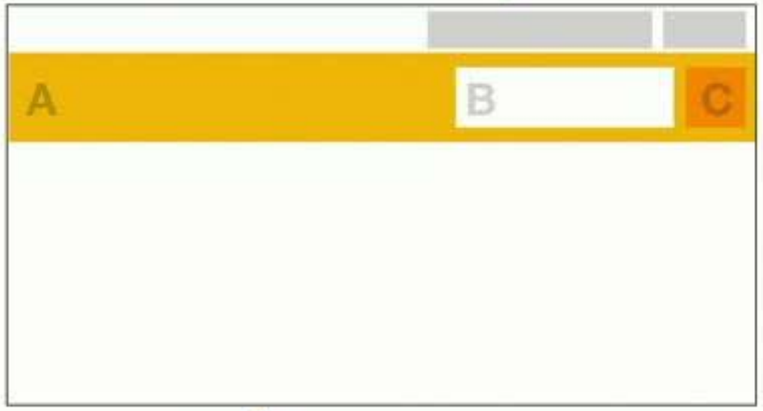
Symbolic

Symbolic

Browser

Rendering

Symbolic



Sizes & positions depend on W, H, Z

Symbolic Web Browser

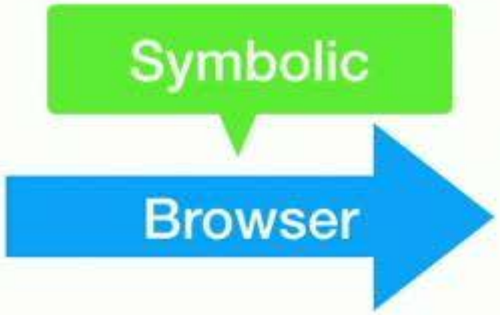
Web Page

HTML + CSS

Configuration

Size: $W \times H$

Text zoom: $Z\times$



Rendering

Problem: write a symbolic web browser

Symbolic Web Browser

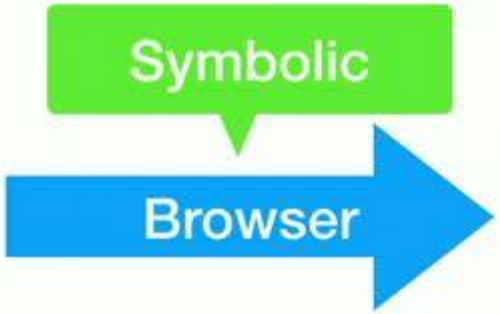
Web Page

HTML + CSS

Configuration

Size: $W \times H$

Text zoom: $Z\times$



Rendering

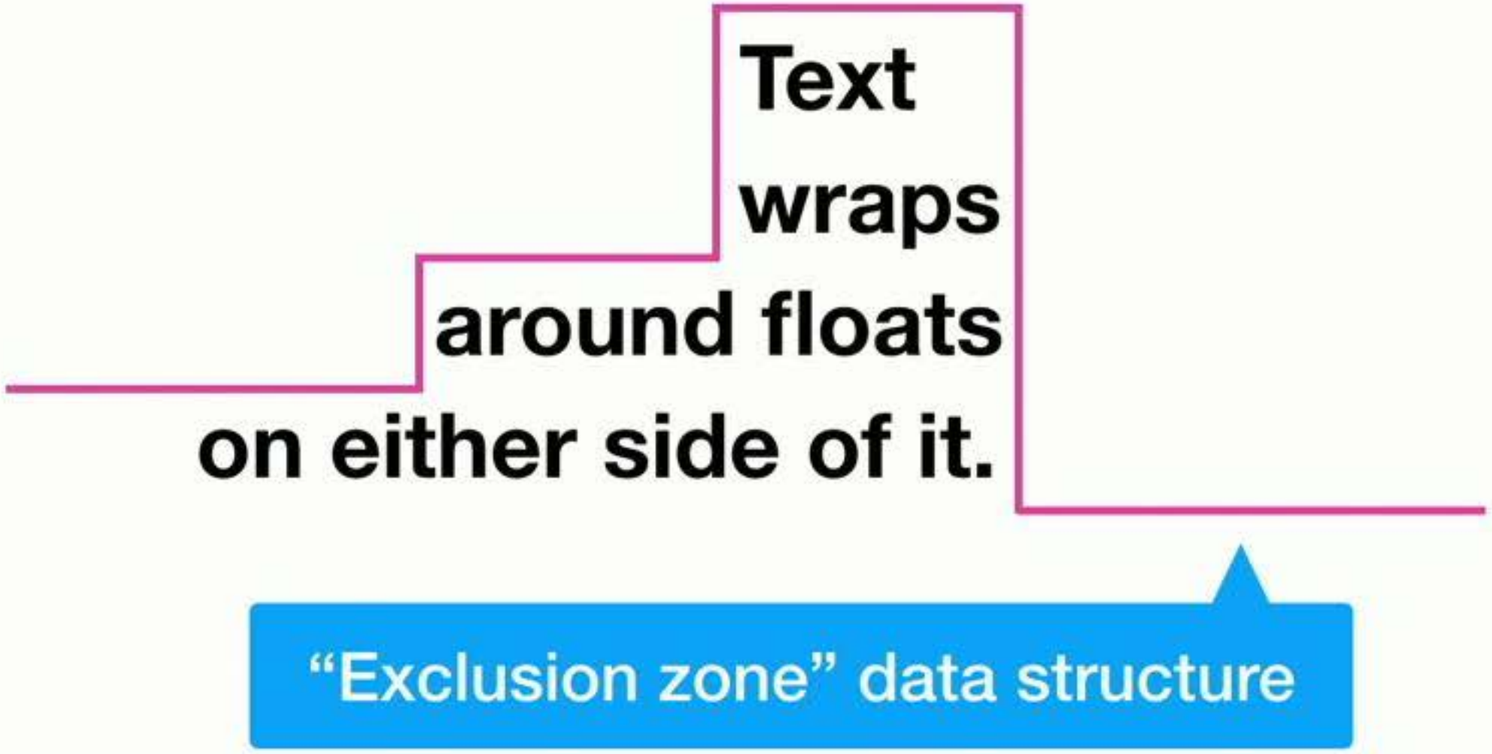
Problem: write a symbolic web browser

Encodings for visual properties

Scaling symbolic reasoning

Encoding Visual Props

Encoding Visual Props



Encoding Visual Props

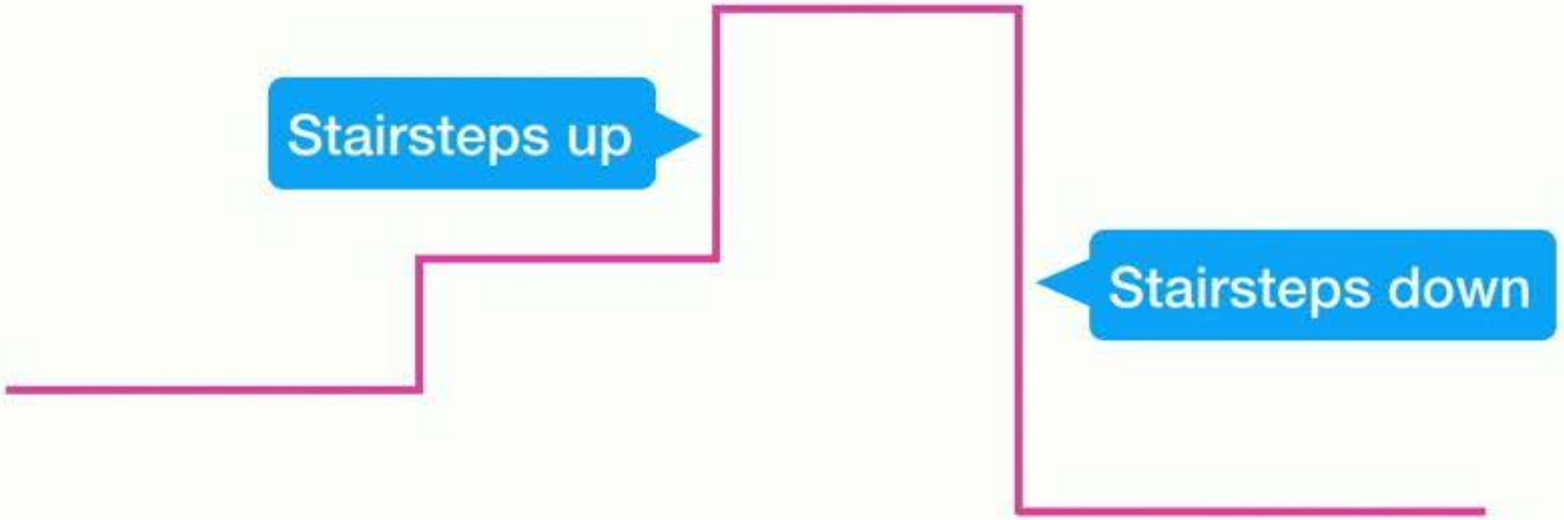
Encoding Visual Props

ezone.**add**(size, position)

Encoding Visual Props

ezone.**add**(size, position) ezone.**place**(size) → position

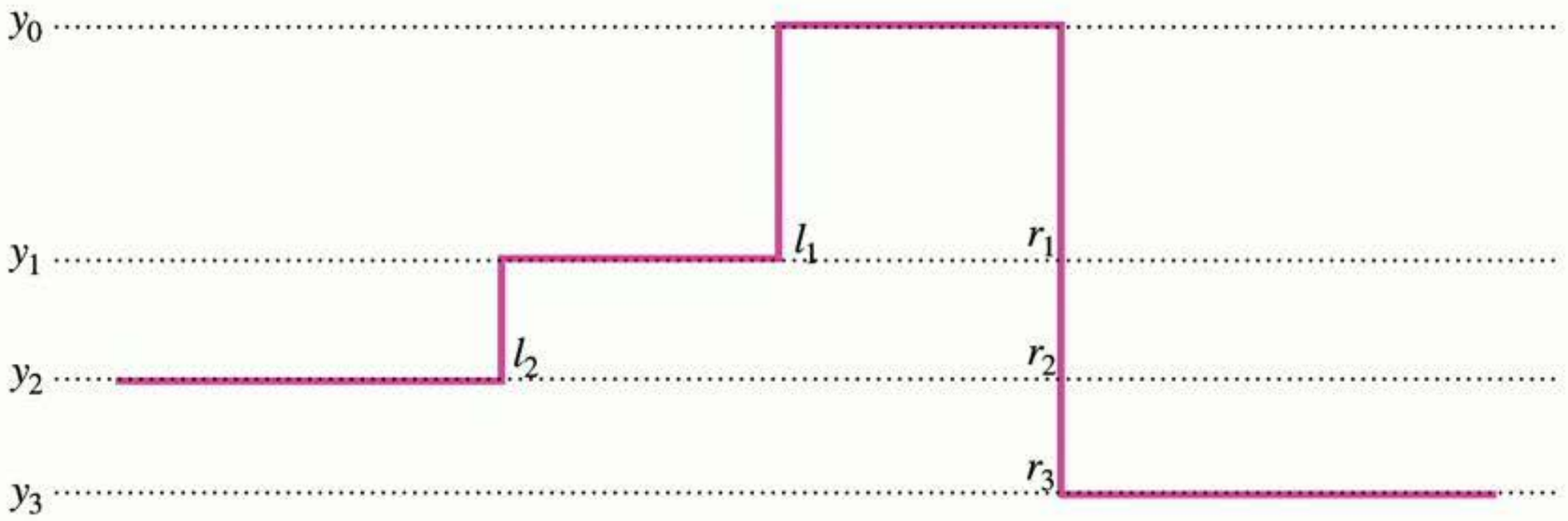
Encoding Visual Props



ezone.**add**(size, position)

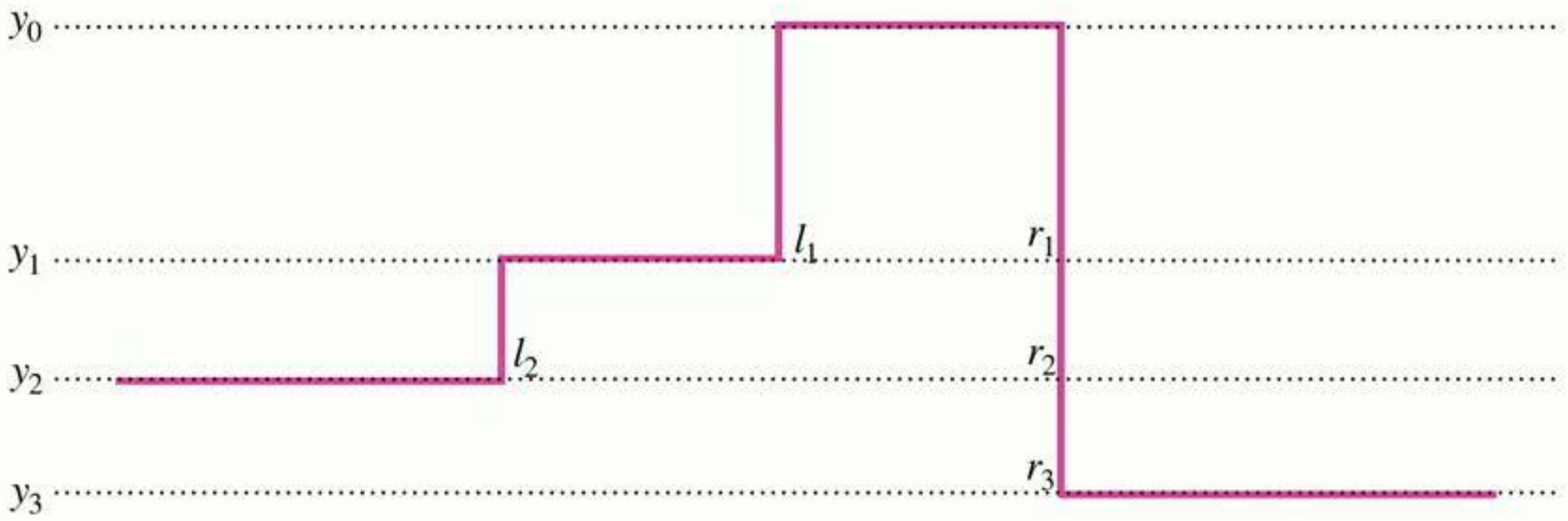
ezone.**place**(size) → position

Encoding Visual Props



ezone.**add**(size, position) ezone.**place**(size) → position

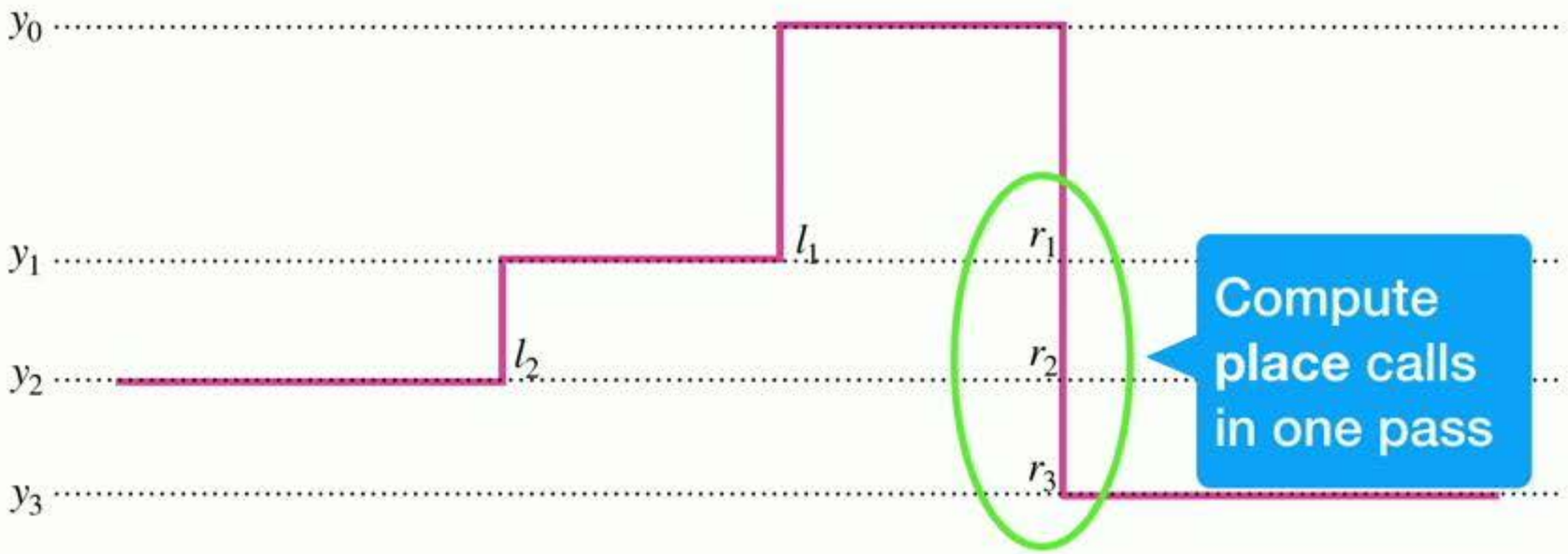
Encoding Visual Props



ezone = y_0 (y_1, l_1, r_1) (y_2, l_2, r_2) $(y_3, \perp, r_3)$

ezone.**add**(size, position) ezone.**place**(size) → position

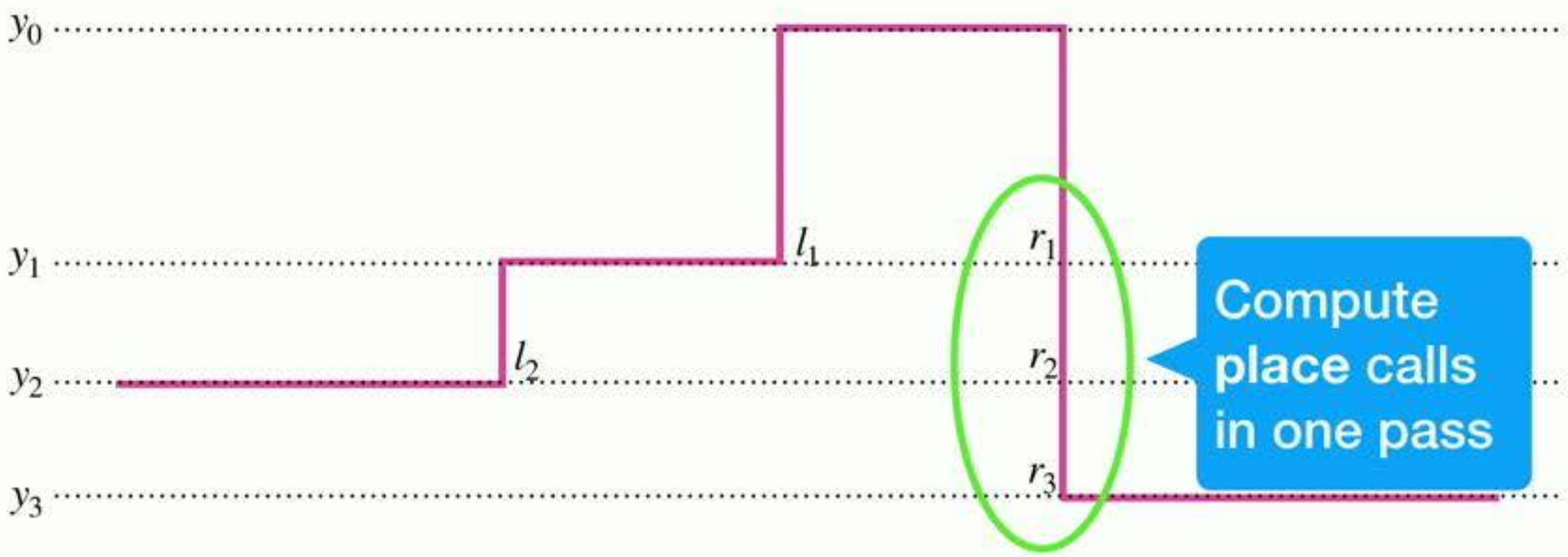
Encoding Visual Props



ezone = y_0 (y_1, l_1, r_1) (y_2, l_2, r_2) $(y_3, \perp, r_3)$

ezone.**add**(size, position) ezone.**place**(size) → position

Encoding Visual Props



ezone = y_0 (y_1, l_1, r_1) (y_2, l_2, r_2) $(y_3, \perp, r_3)$

ezone.**add**(size, position)

O(n) equations

ezone.**place**(size) → position

O(n) equations

Validating the Formalization

Validating the Formalization

Standard Tests

Validating the Formalization

Non-automated

Standard Tests

Validating the Formalization

Non-automated

Standard Tests

Browsers

Validating the Formalization

Non-automated

Standard Tests

Browsers

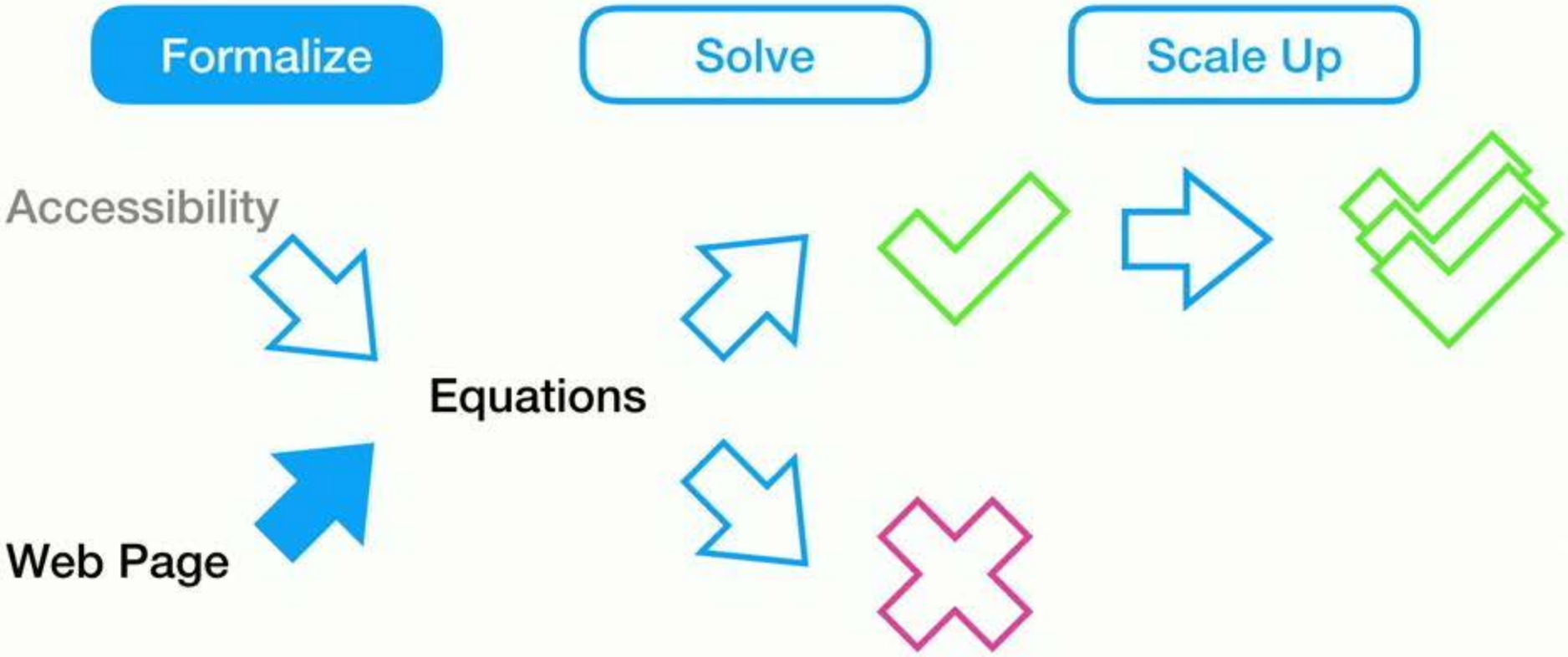
Evaluate: Differential testing vs browsers

[OOPSLA'16]

Pass thousands of conformance tests

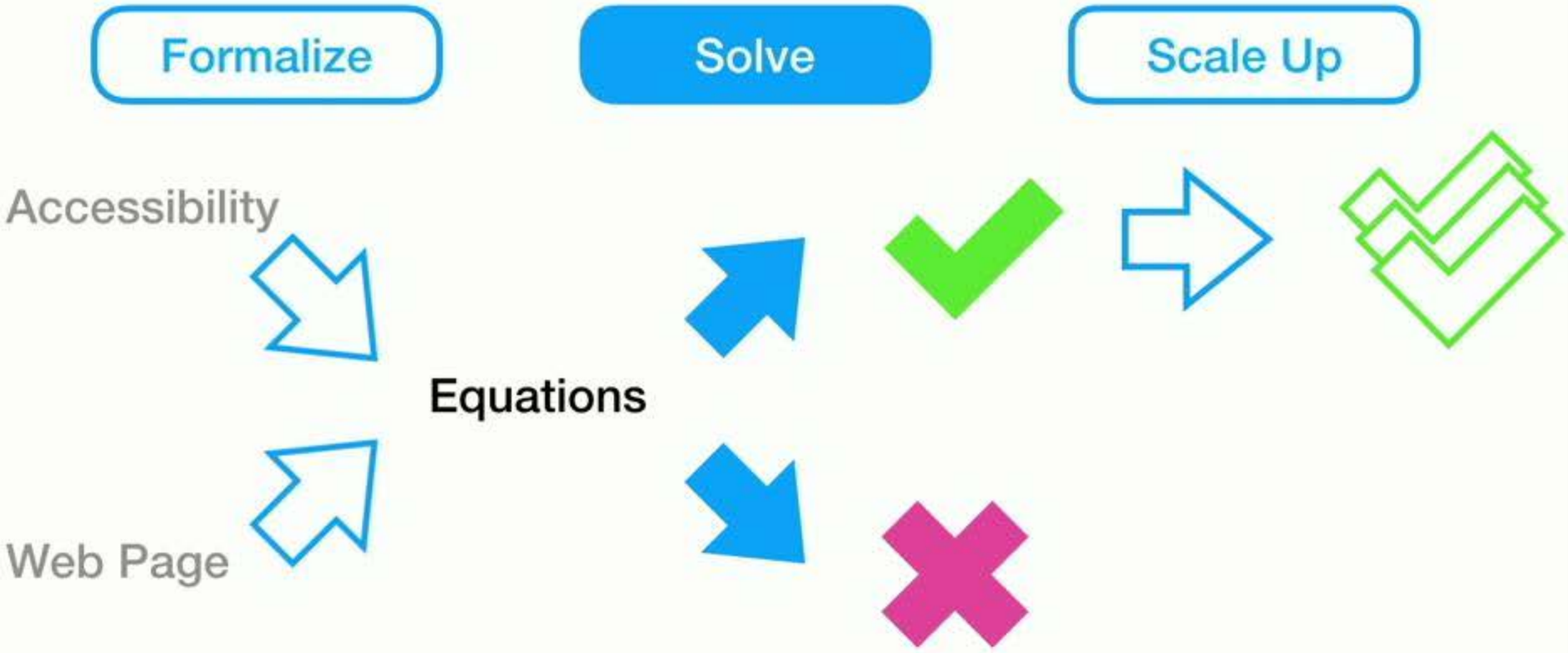
Found bugs in existing browsers

How VizAssert Works



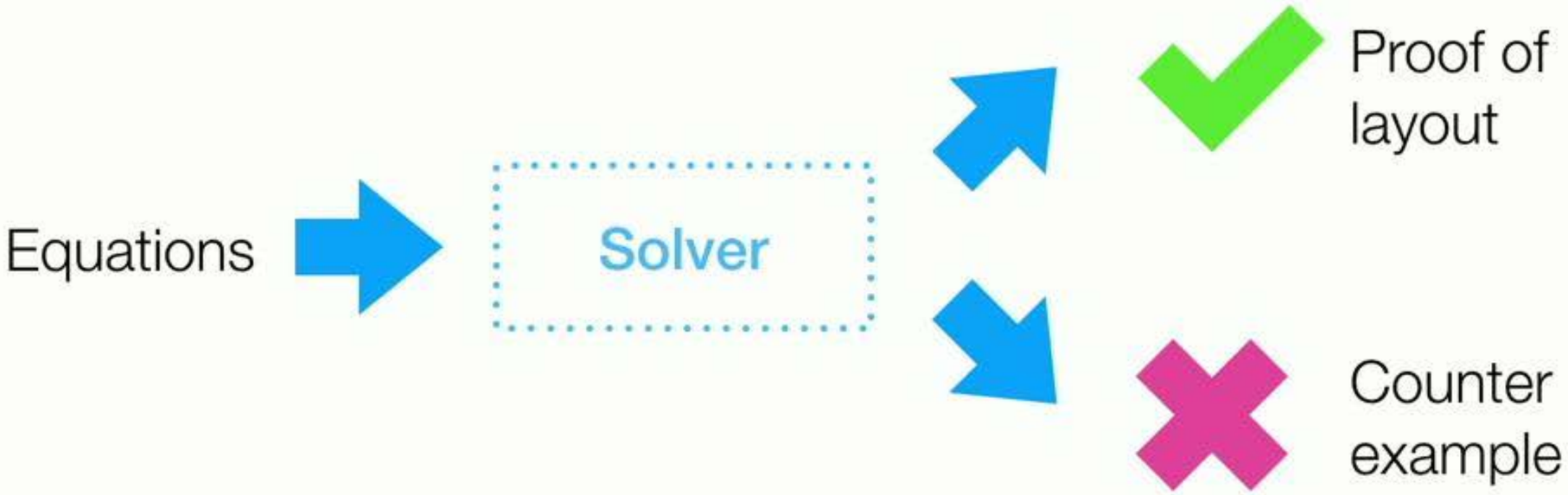
Formalizing visual properties

How VizAssert Works



Formalizing visual properties

Solving the Equations



Solving the Equations

Solving the Equations

Quantifier **F**ree

Linear

Real **A**rithmetic

Solving the Equations

Quantifier **F**ree

Linear

Real **A**rithmetic ⟶ No rounding error

Solving the Equations

Quantifier **F**ree

Linear ⟶ No multiplication

Real **A**rithmetic ⟶ No rounding error

Solving the Equations

Quantifier **F**ree ⟶ No nested loops

Linear ⟶ No multiplication

Real **A**rithmetic ⟶ No rounding error

Solving the Equations

Simulated Rounding Error

$a = b \longrightarrow -\varepsilon < a - b < \varepsilon$

Real **A**rithmetic ⟶

No nested loops

No multiplication

No rounding error

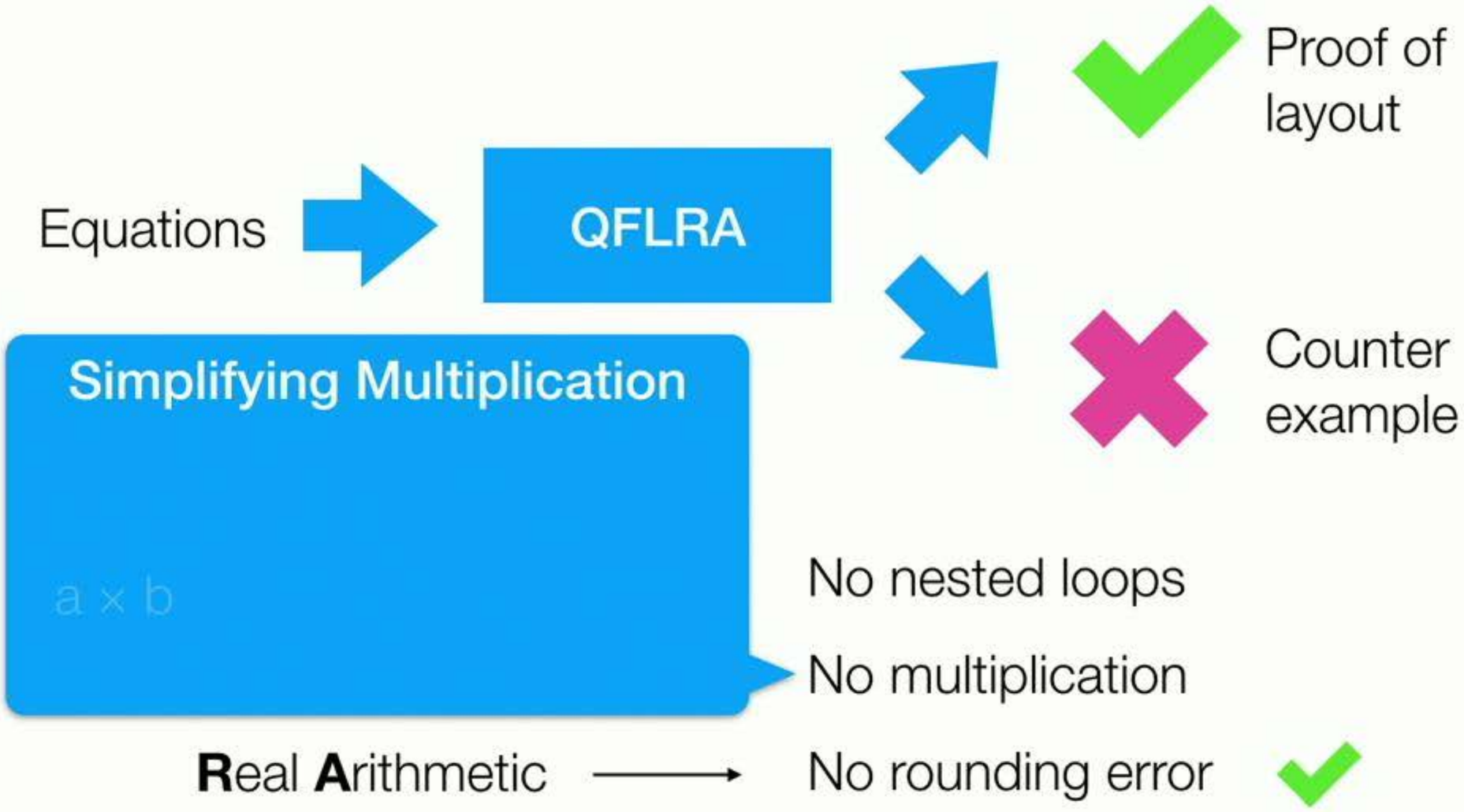
Solving the Equations

Quantifier **F**ree ⟶ No nested loops

Linear ⟶ No multiplication

Real **A**rithmetic ⟶ No rounding error ✓

Solving the Equations



Solving the Equations

Quantifier **F**ree ⟶ No nested loops

Linear ⟶ No multiplication ✓

Real **A**rithmetic ⟶ No rounding error ✓

Solving the Equations

Bounded work per element

Only one loop: render each element

Quantifier **F**ree → No nested loops

Linear → No multiplication ✓

Real **A**rithmetic → No rounding error ✓

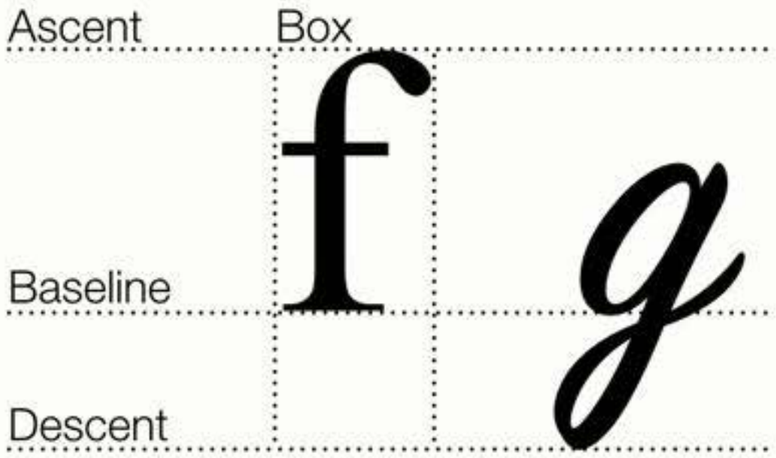
Solving the Equations

Problem: bounded work per element

Incrementalization, fusion, and unrolling

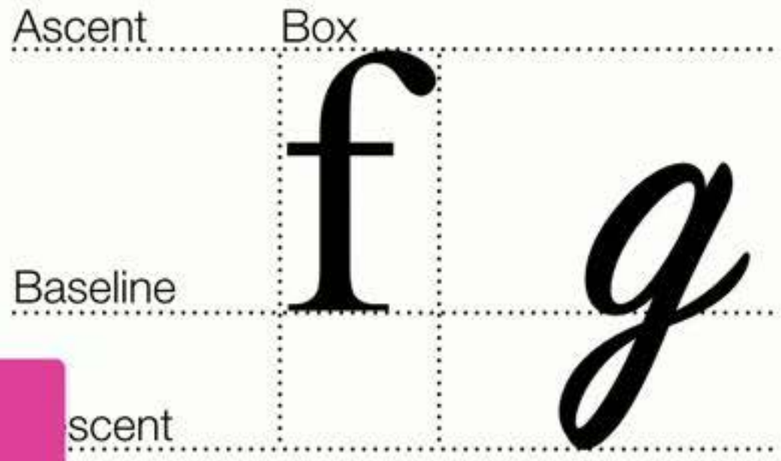
Inspired by compiler optimizations

Computing Line Height



"The line box height is the distance between the **uppermost** box top and the **lowermost** box bottom"

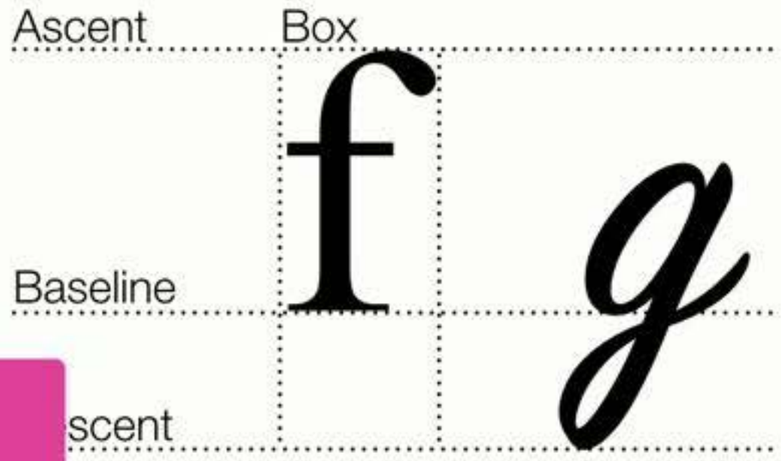
Computing Line Height



Loop over elements in line

"The line box height is the distance between the **uppermost** box top and the **lowermost** box bottom"

Computing Line Height

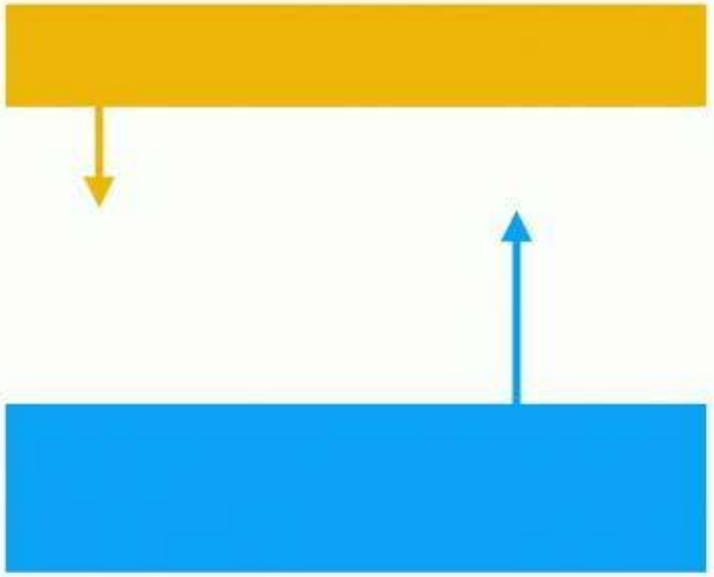


Loop over elements in line

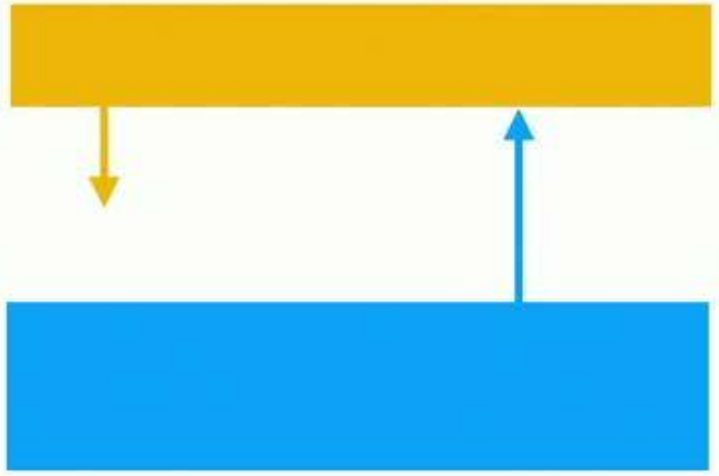
"The line box height is the distance between the **uppermost** box top and the **lowermost** box bottom"

Incrementalization: update running maximum

Computing Margins

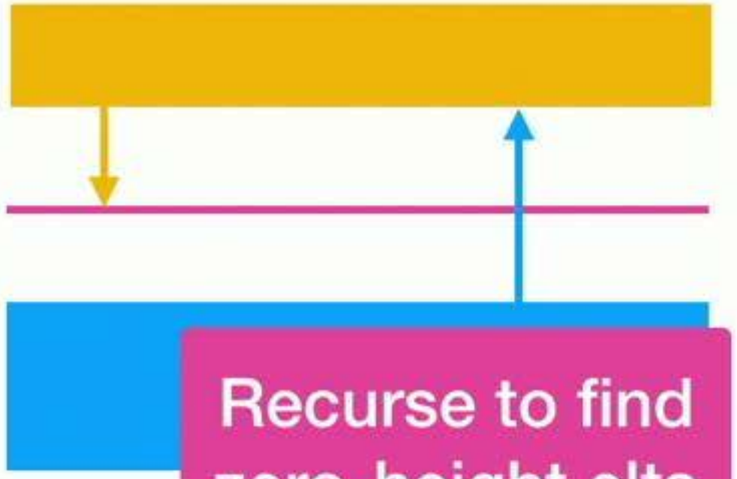


Computing Margins



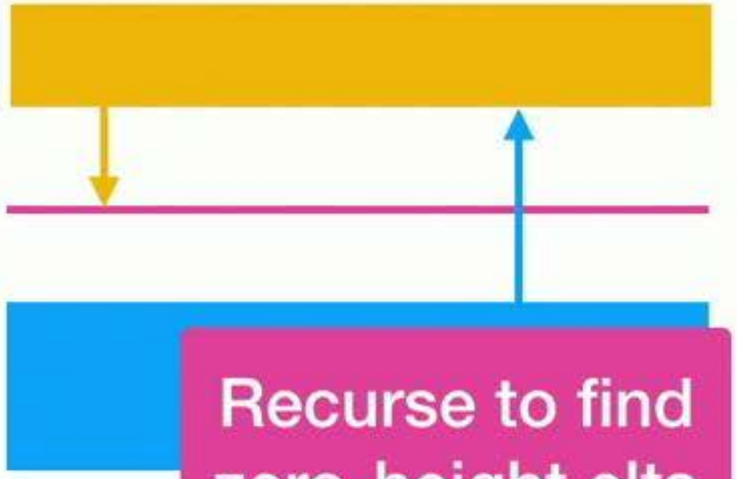
"Adjoining vertical margins **collapse**;
two margins are **adjoining** if and only if …"

Computing Margins



"Adjoining vertical margins **collapse**;
two margins are **adjoining** if and only if …"

Computing Margins



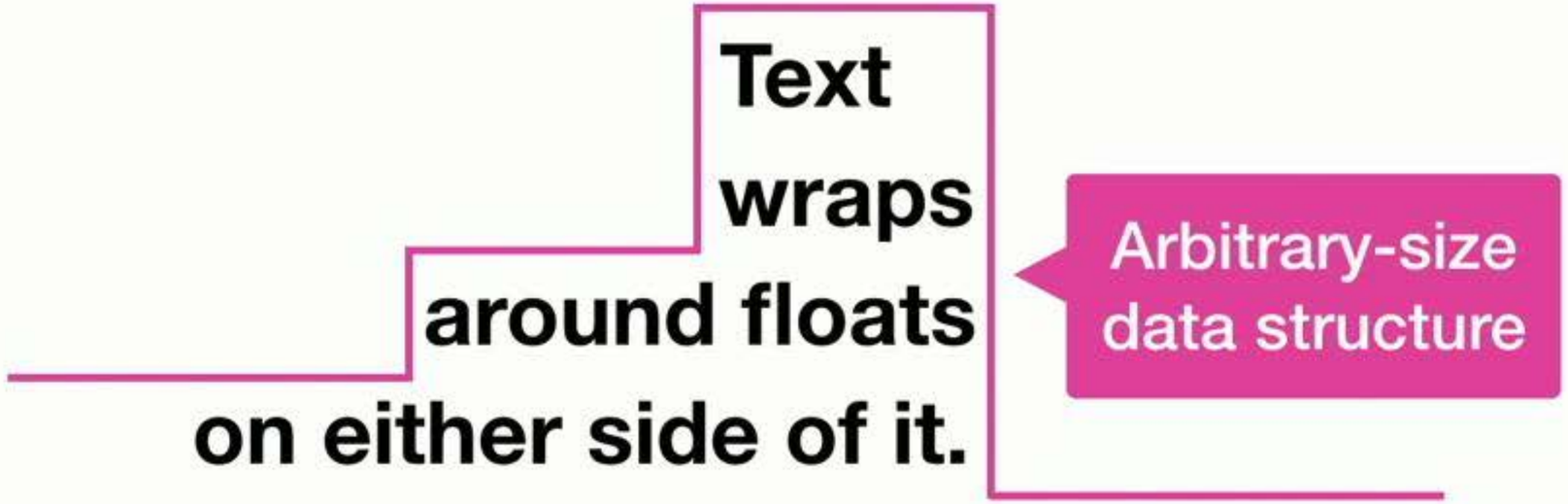
Recurse to find zero-height elts

"Adjoining vertical margins **collapse**;
two margins are **adjoining** if and only if …"

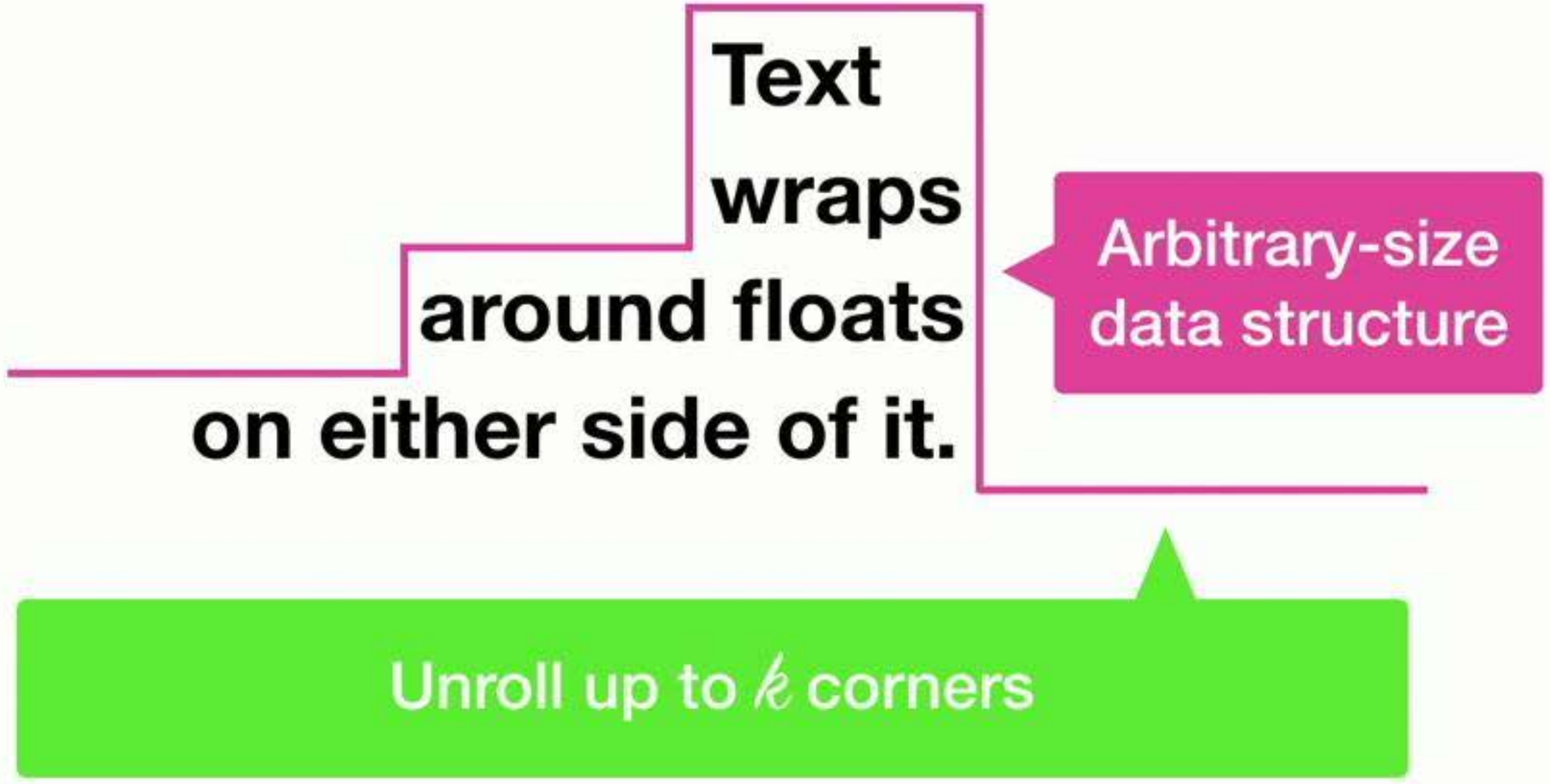
Fusion: interleave with outer render loop

Computing Float Layout

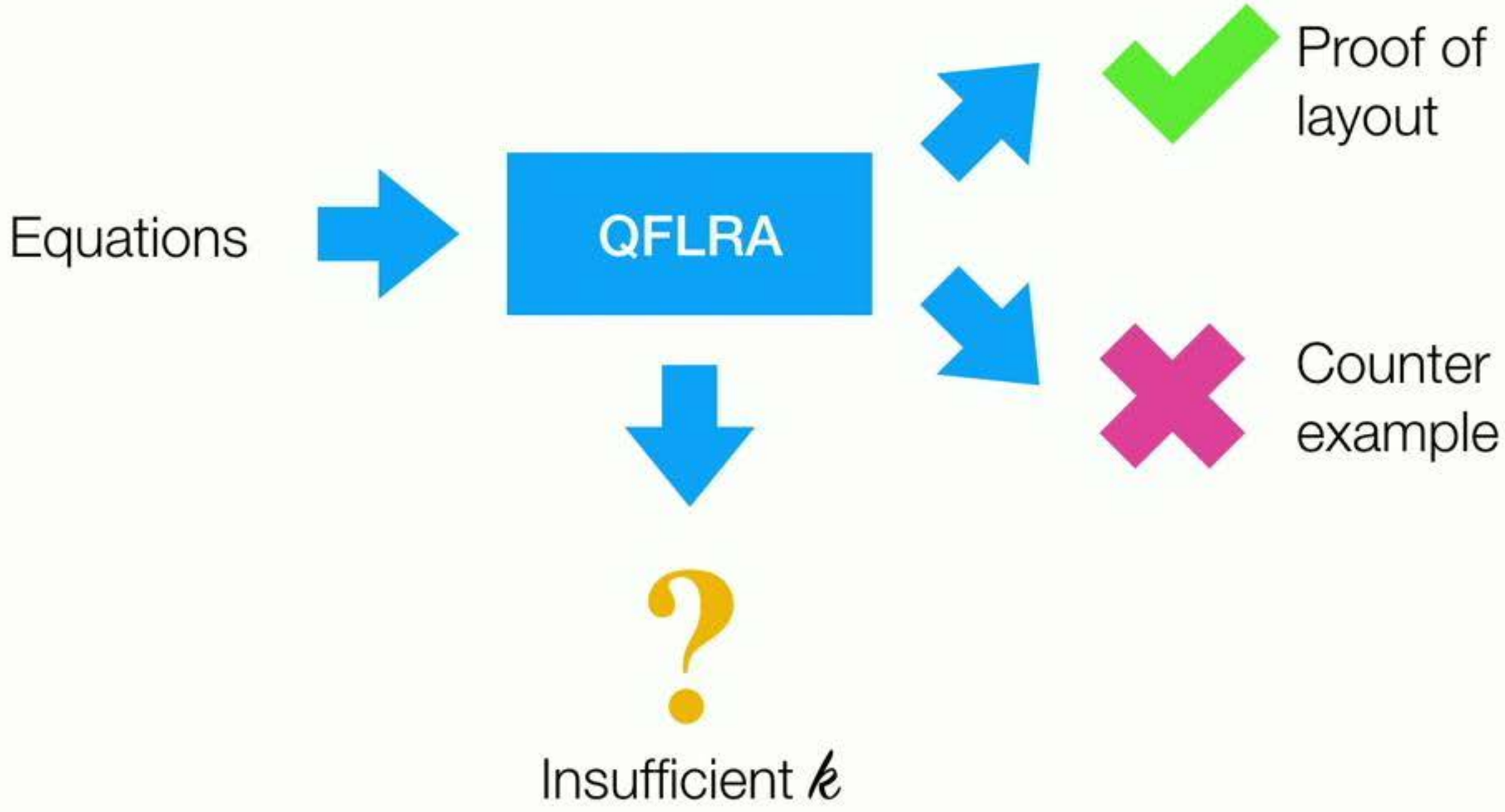
Computing Float Layout



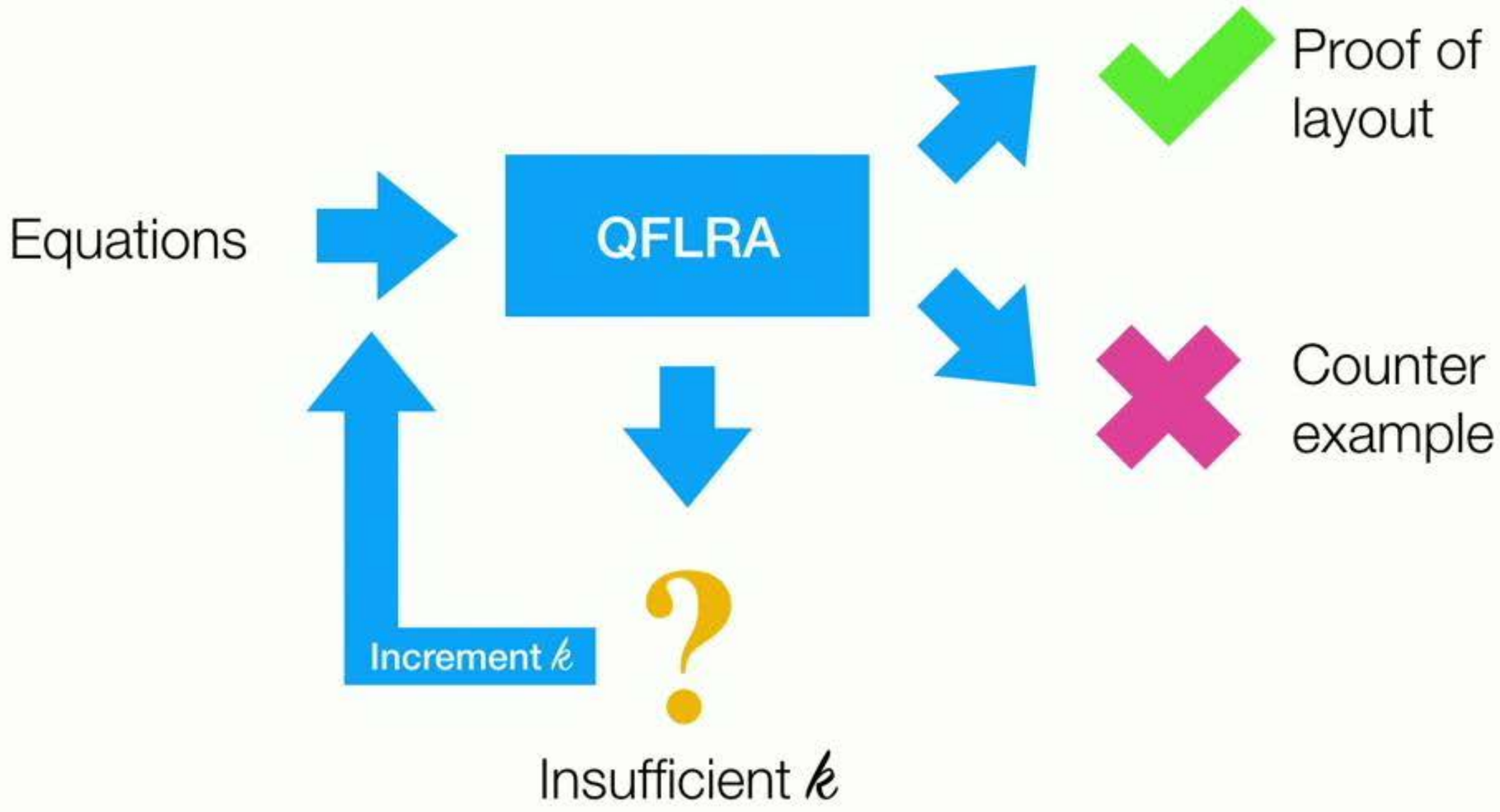
Computing Float Layout



Solving the Equations



Solving the Equations



Verifying Real Pages

Websites from design forum FreeWebsiteTemplates.com

Verifying Real Pages

Webs lates.com

Only use formalized subset of CSS

62 web pages

Verifying Real Pages

Webs lates.com

Only use formalized subset of CSS

62 web pages

14 properties

From Apple, Google, DoD accessibility guides

Verifying Real Pages

Webs lates.com

Only use formalized subset of CSS

62 web pages

14 properties

From Apple, Google, DOJ accessibility guides

Verifying Real Pages

Webs lates.com

Only use formalized subset of CSS

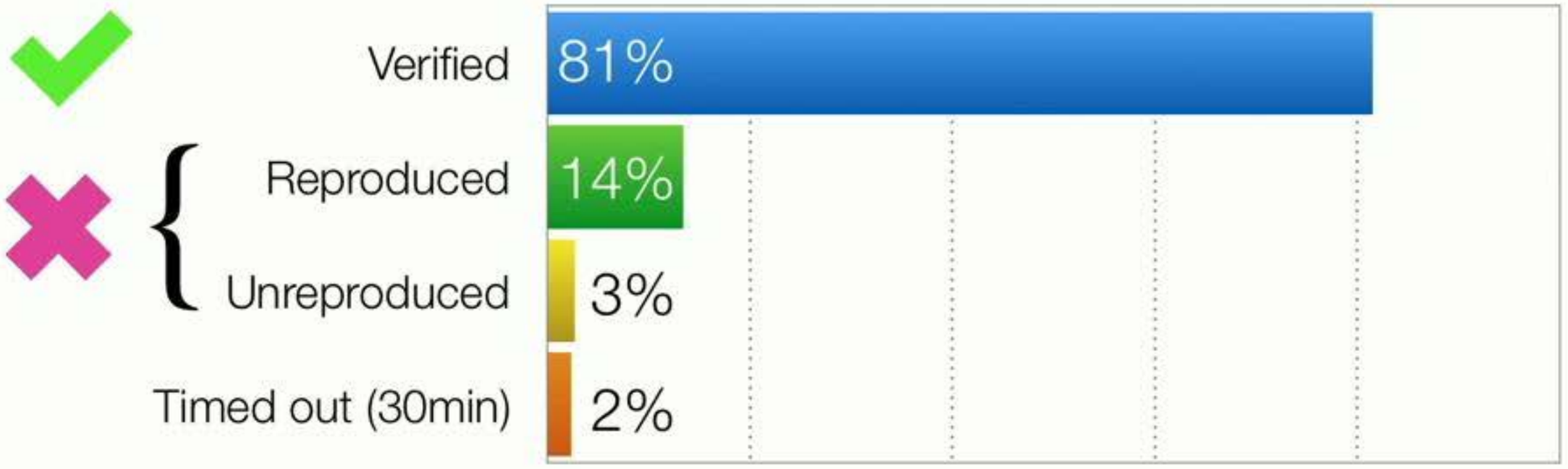
62 web pages

14 properties

} **476** sensible combinations

From Apple, Google, DOJ accessibility guides

Verifying Real Pages



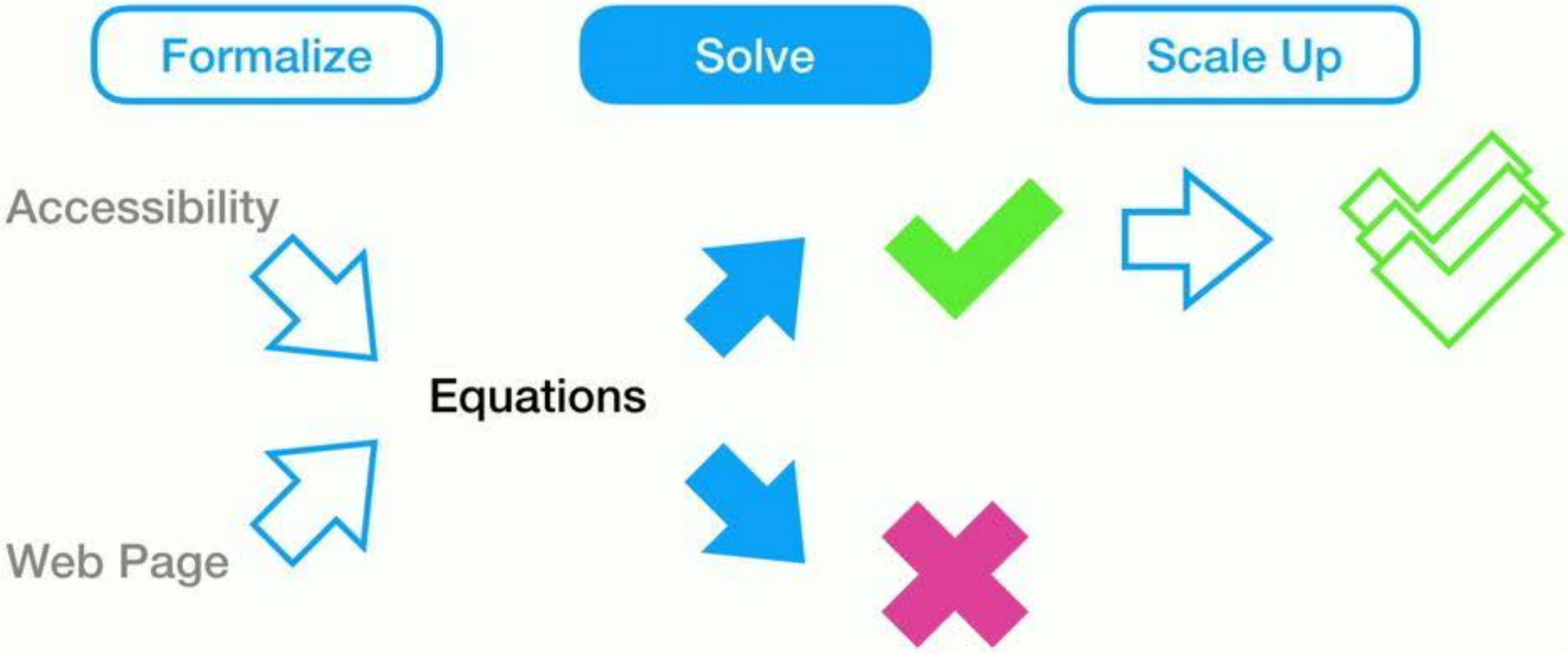
Evaluate: verified majority of real-world inputs

Found many real accessibility bugs

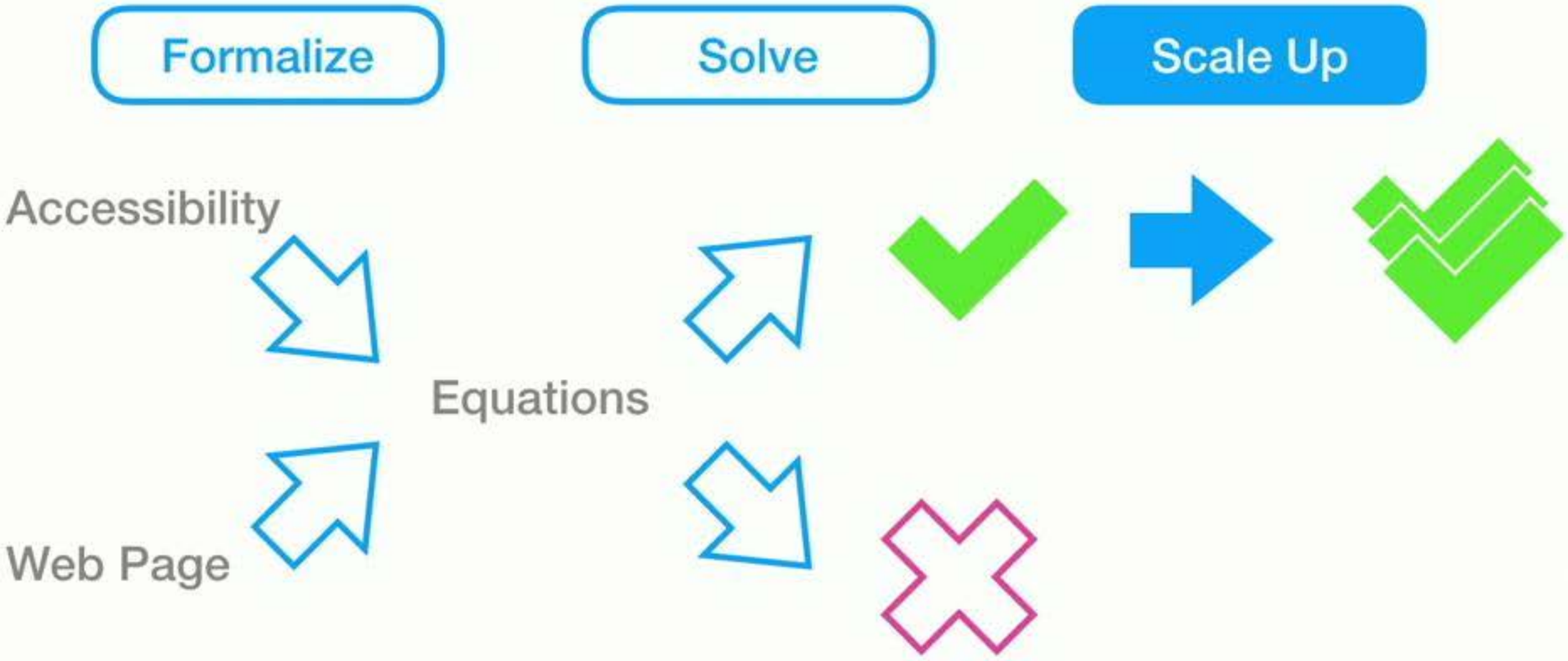
Few false positives and few timeouts

[PLDI'18]

How VizAssert Works



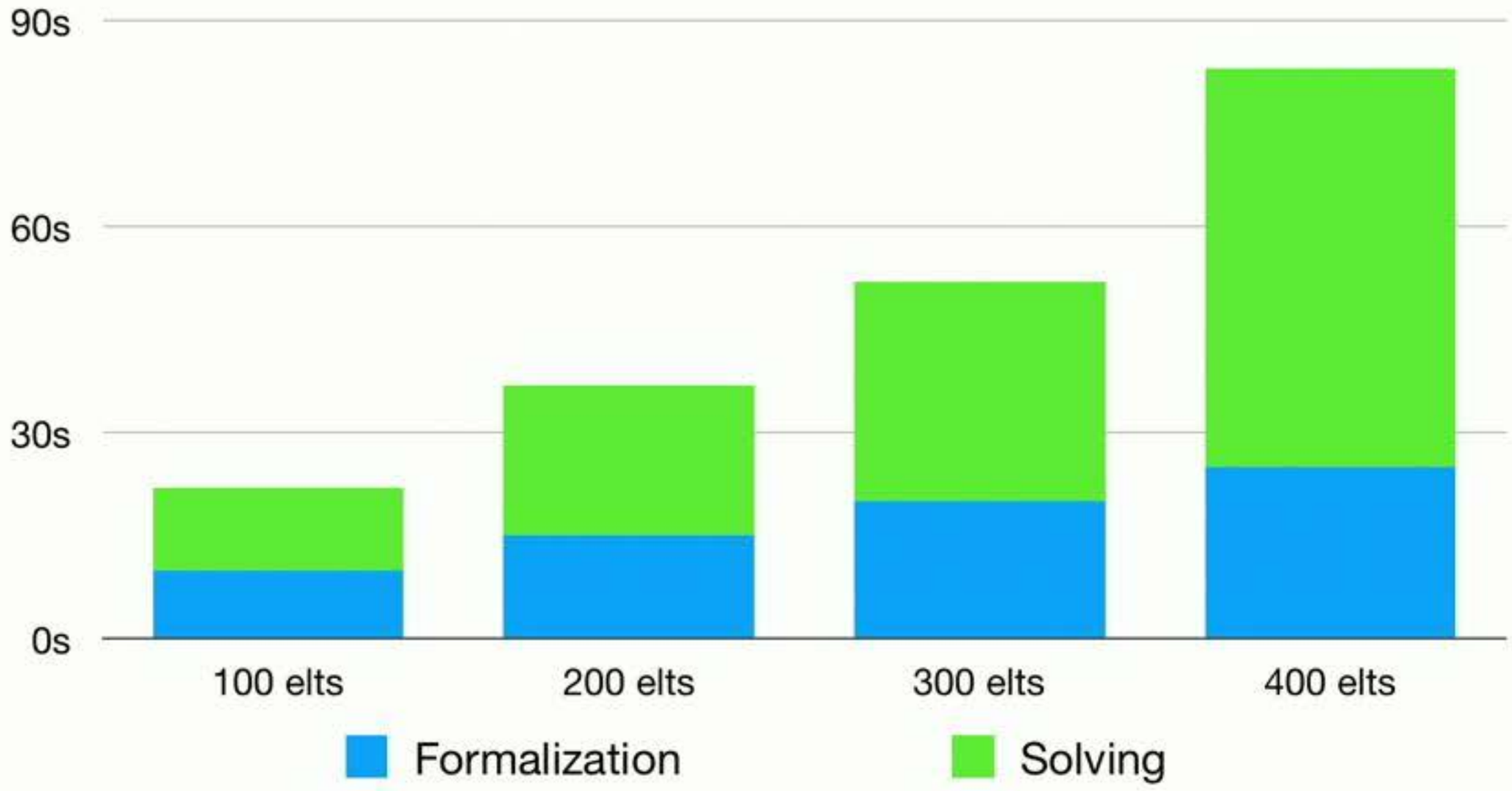
How VizAssert Works



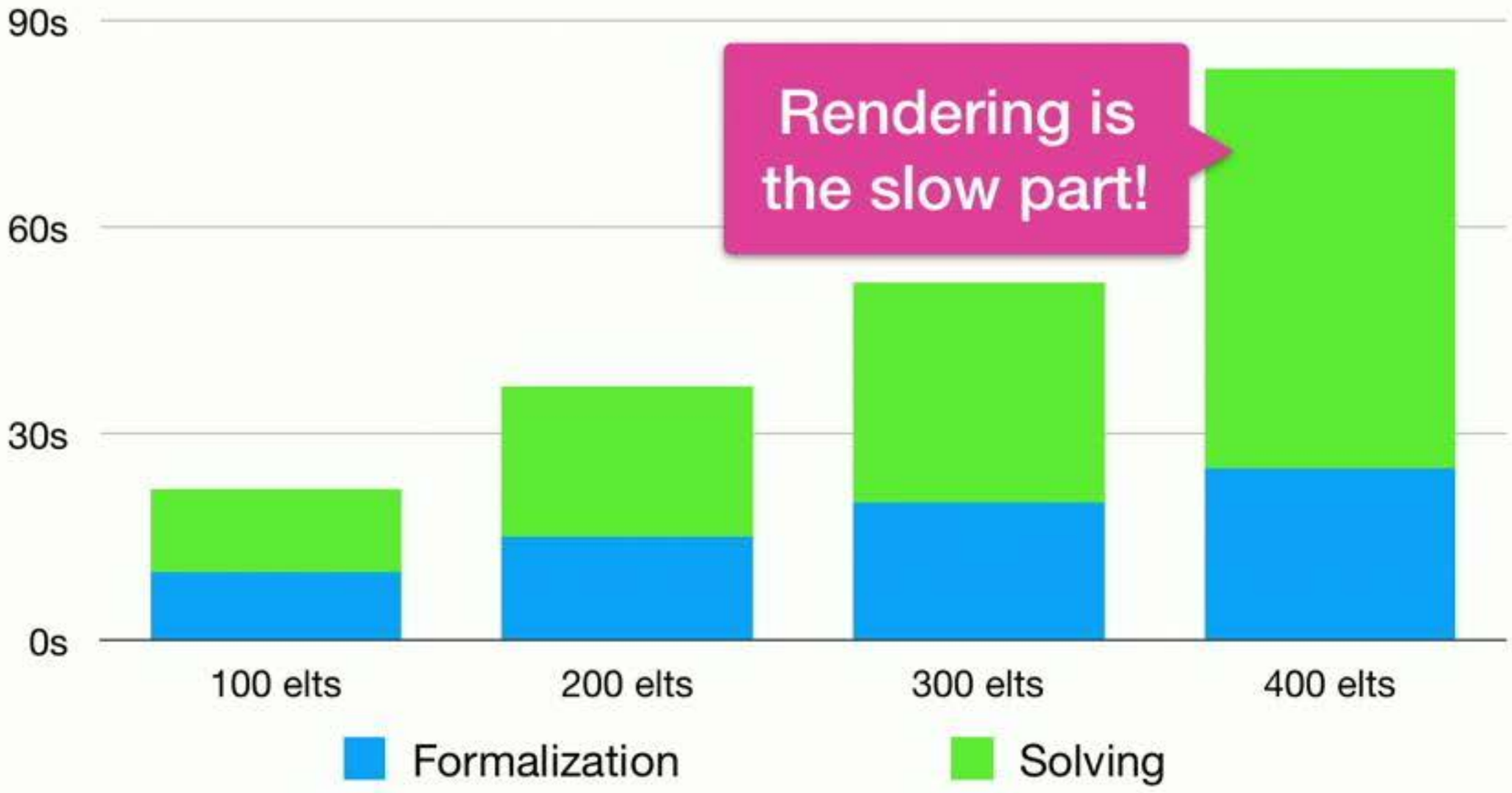
Formalizing visual properties

Overcoming solver limitations

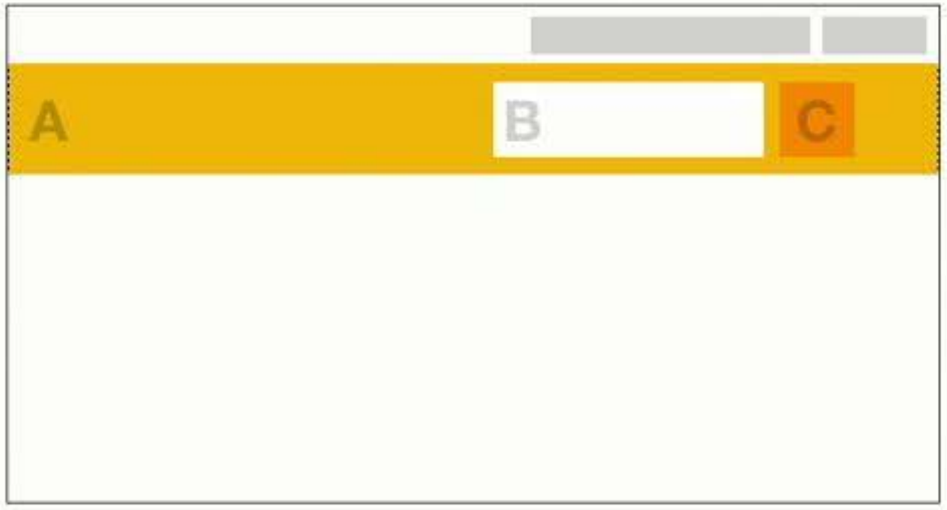
Scaling Verification



Scaling Verification



Scaling Verification



Problem: reason about large pages quickly

Scaling Verification

Problem: reason about large pages quickly

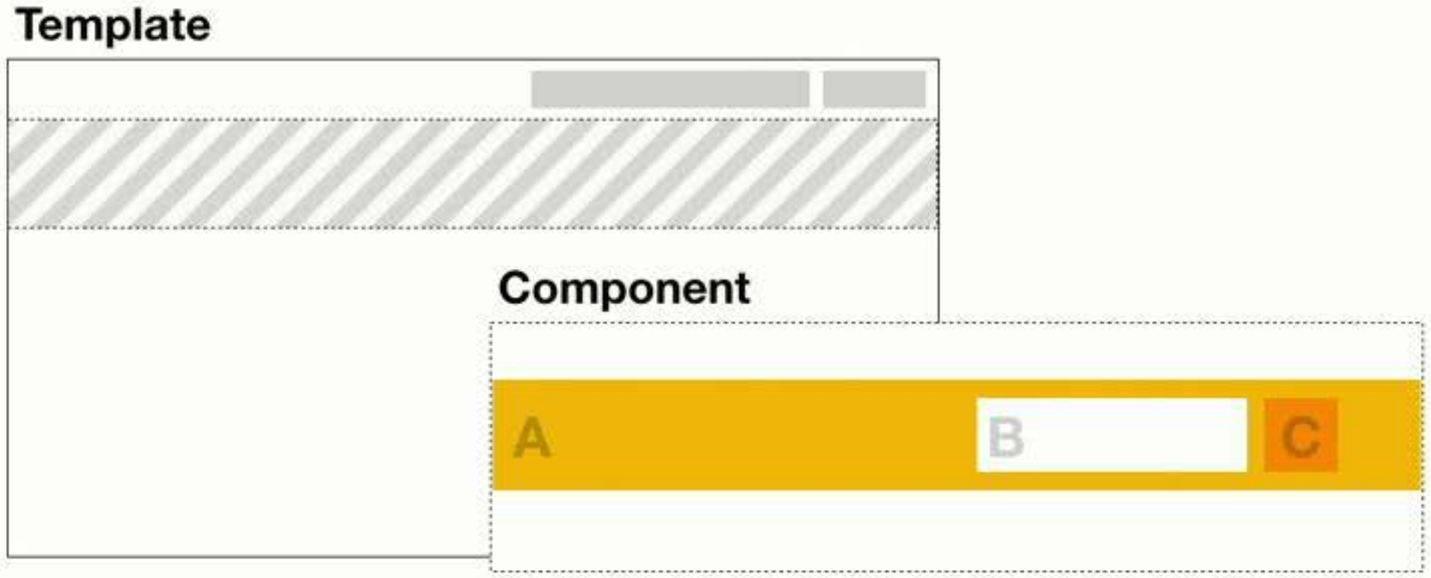
Scaling Verification

Problem: reason about large pages quickly

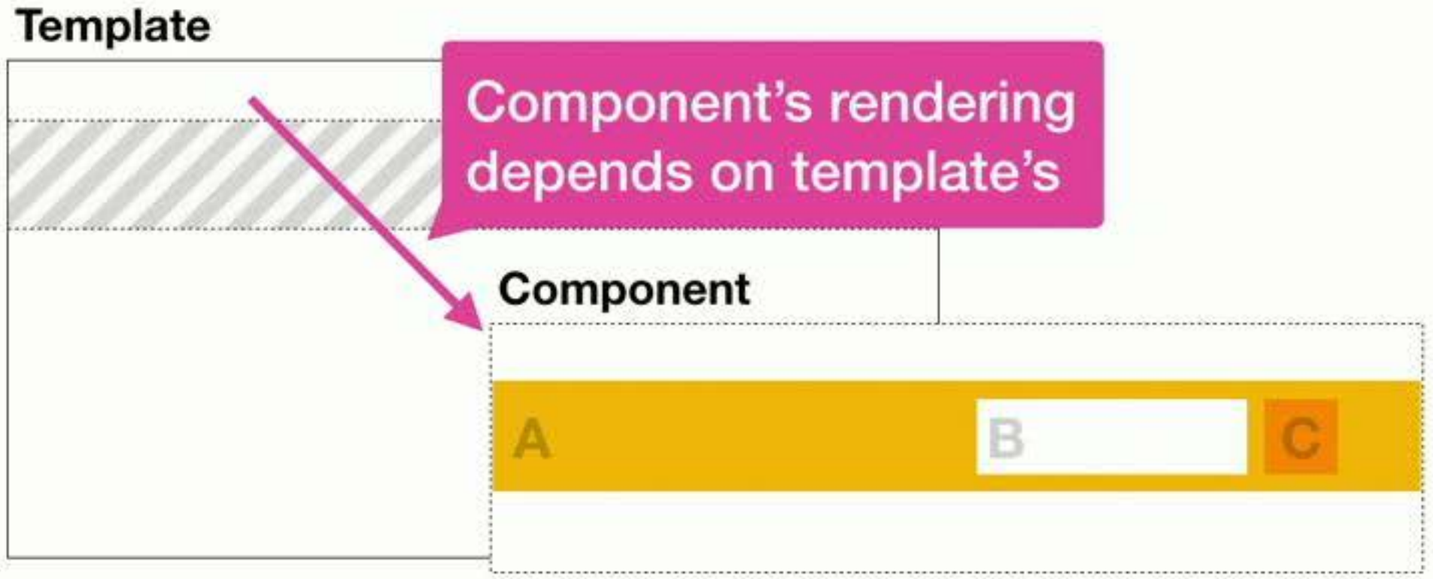
Divide web page into small components

Combine components with rely/guarantee logic

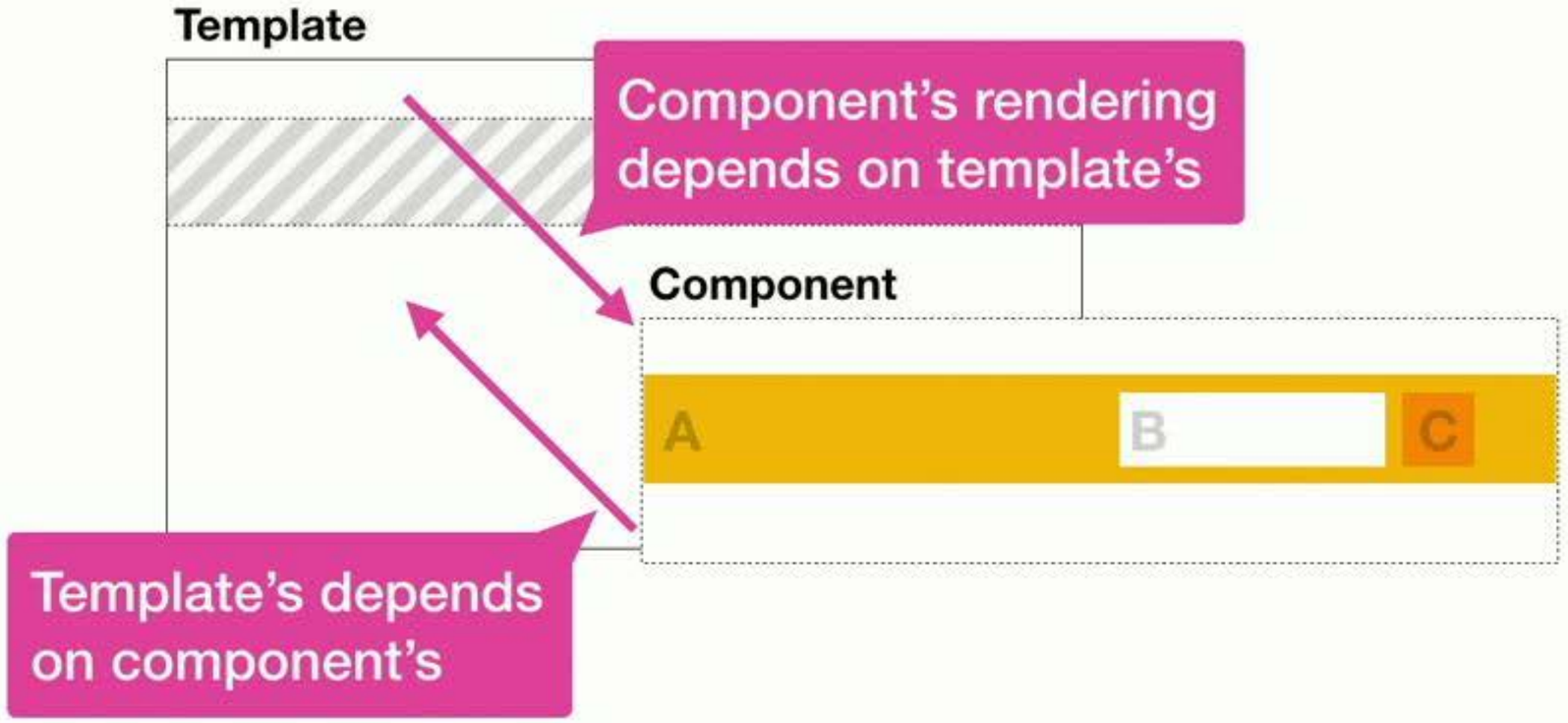
Isolating Components



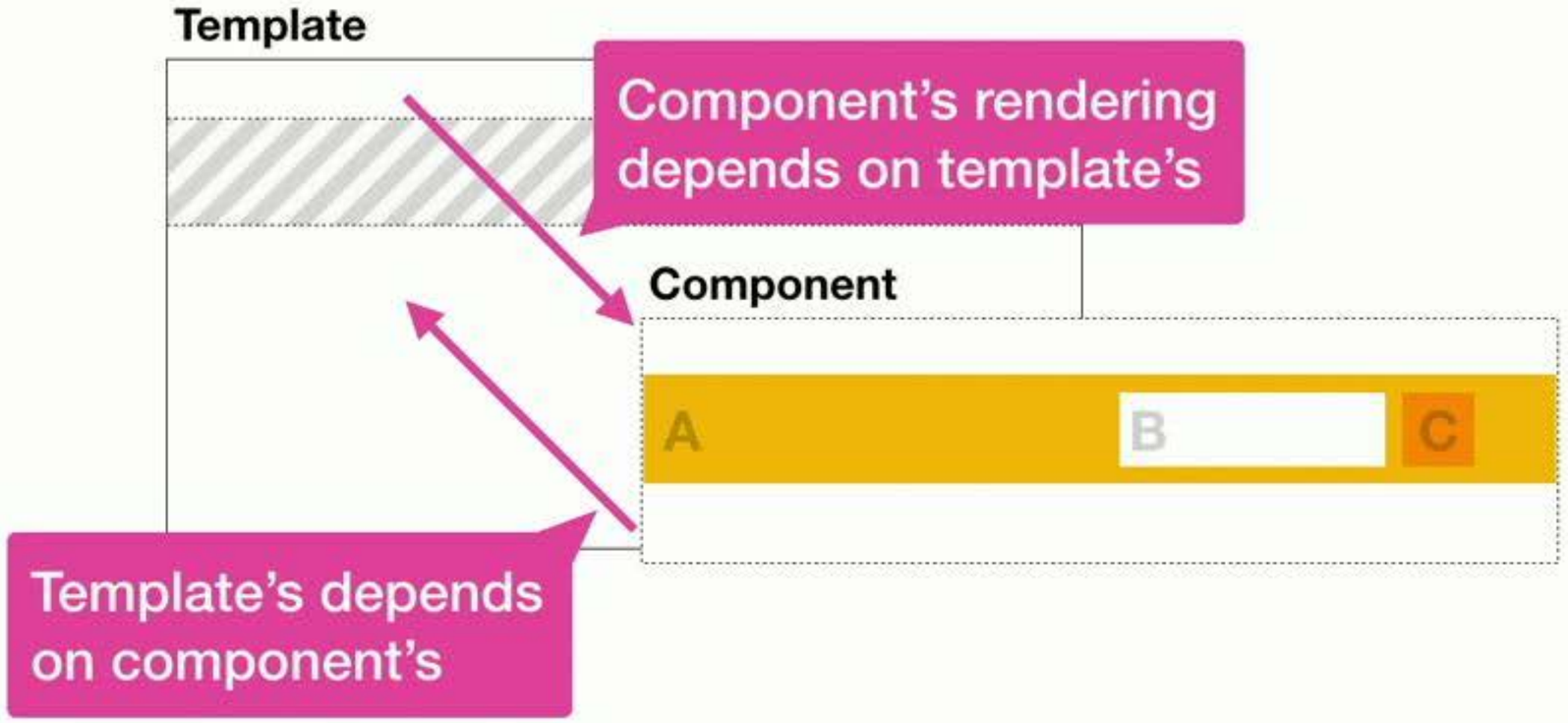
Isolating Components



Isolating Components



Isolating Components

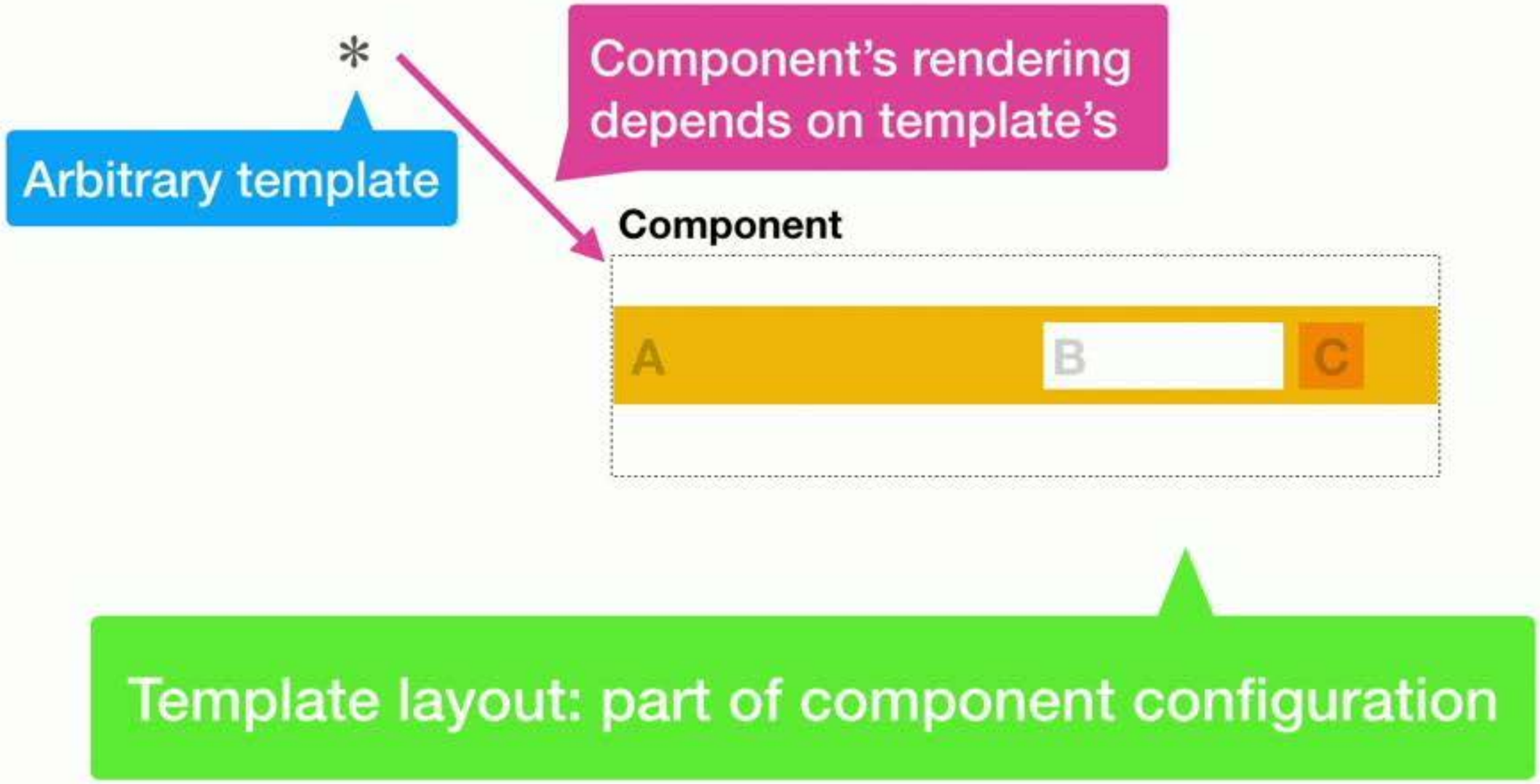


No module or function boundaries!

Isolating Components

No module or function boundaries!

Isolating Components



Isolating Components

Precondition

Width available
Current font size
Floating elements

Component's rendering depends on template's

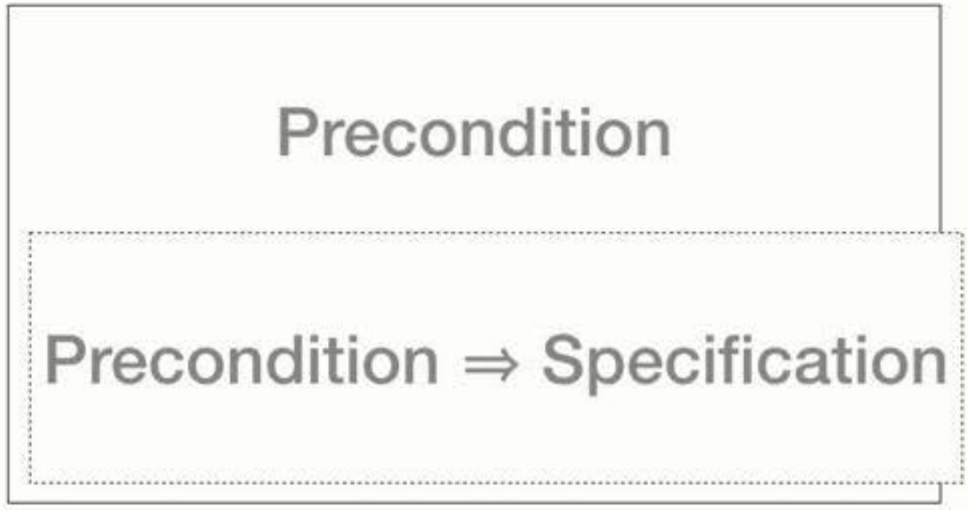
Template layout: part of component configuration

Rely / Guarantee

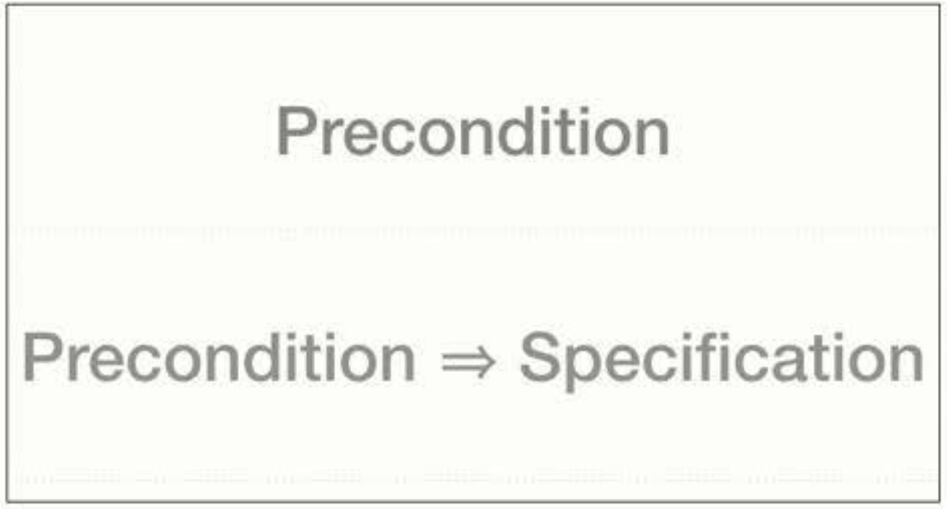
Component

Precondition ⇒ Specification

Rely / Guarantee



Rely / Guarantee



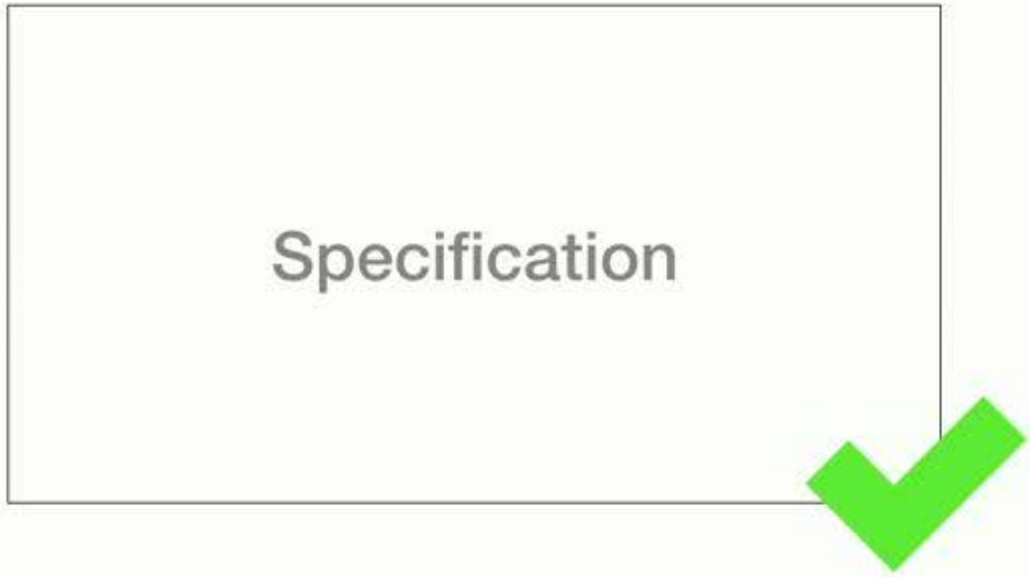
Rely / Guarantee

Precondition

&&

Precondition ⇒ Specification

Rely / Guarantee

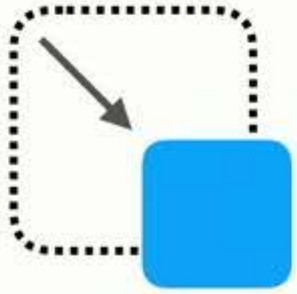


Preconditions checked purely logically

No rendering ⇒ fast

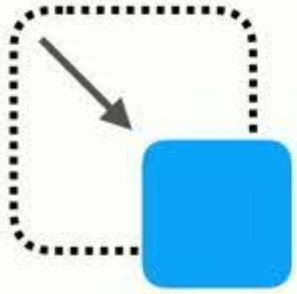
Improving Scale

Problem Size



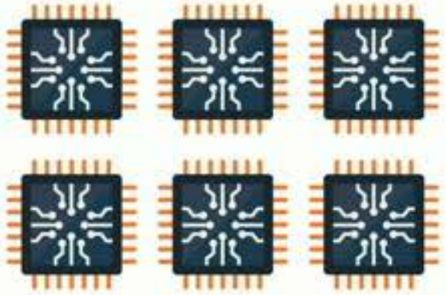
Improving Scale

Problem Size



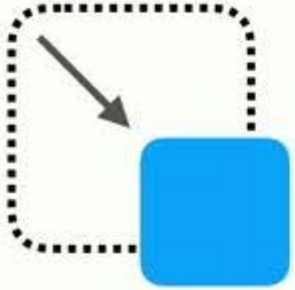
+

Parallelism



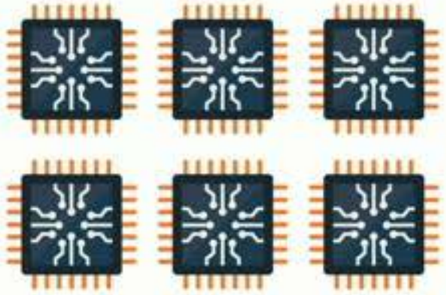
Improving Scale

Problem Size



+

Parallelism

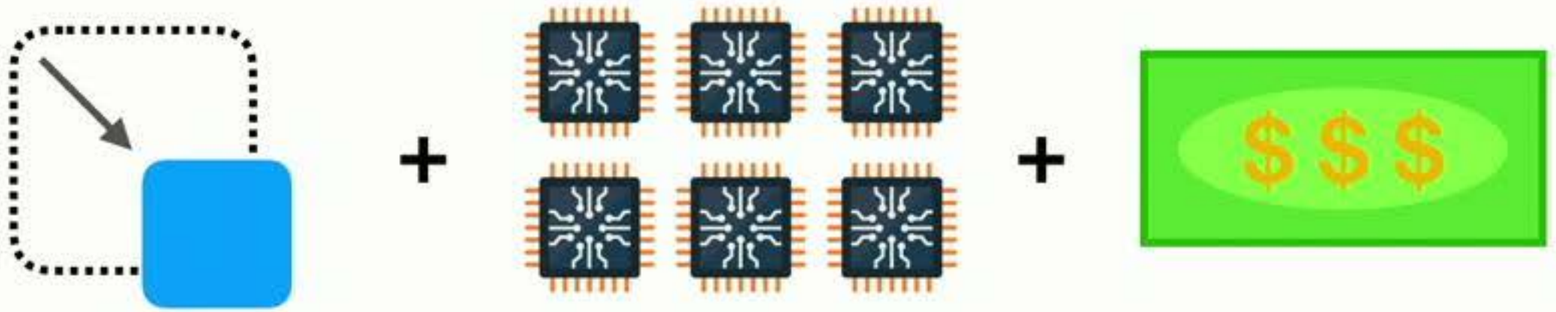


+

Caching

Improving Scale

Problem Size + **Parallelism** + **Caching**

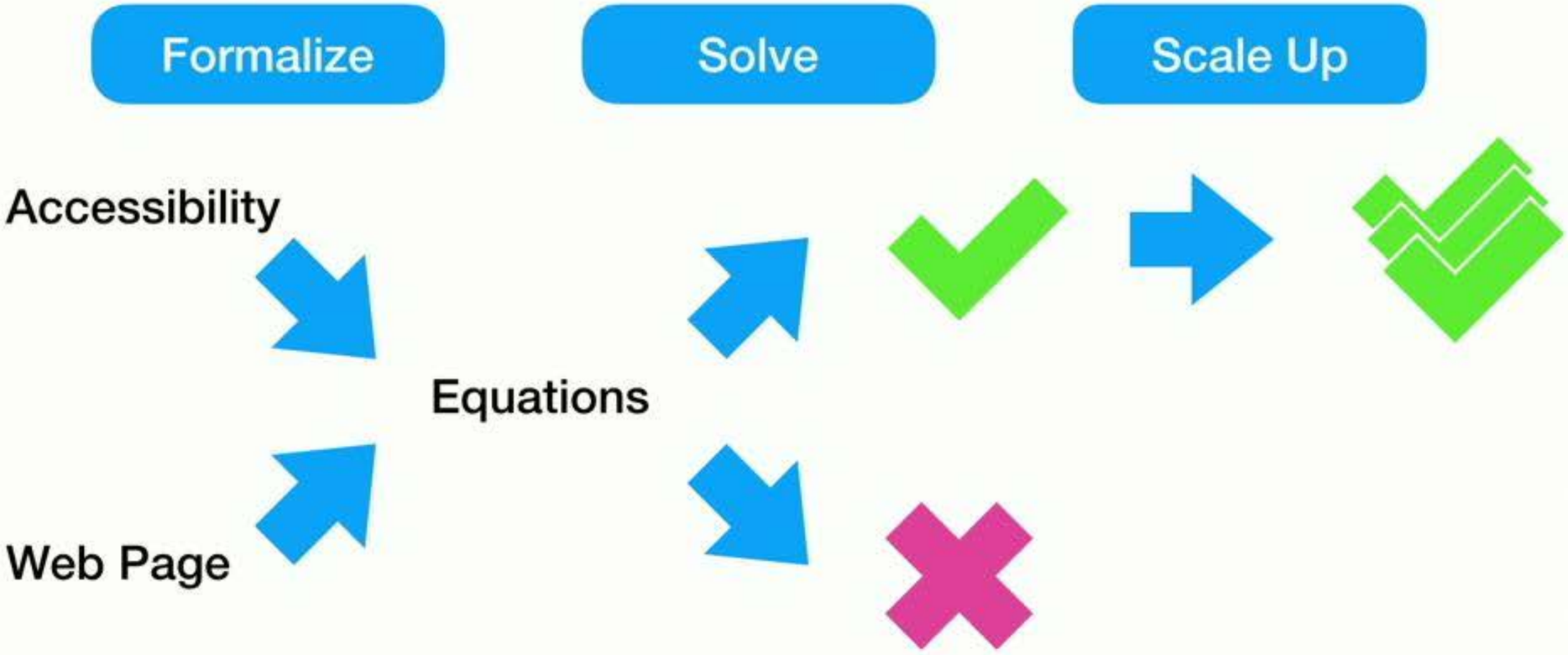


Evaluate: broader class of verified pages

Scaled to 11× larger pages, to 1400× faster

Scaled to multiple pages on one site

How VizAssert Works



Formalizing visual properties

Overcoming solver limitations

Finding modularity for rely/guarantee

Future Work

Frontend code (JavaScript)

Backend code (Python, PHP, …)

Other platforms (Android, iOS…)

User studies with designers

Future Work

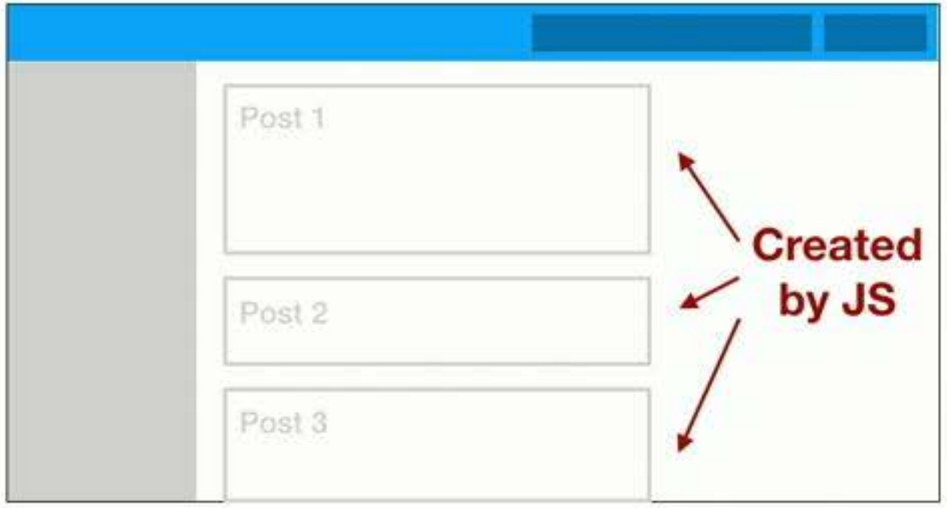
Frontend code (JavaScript)

Backend code (Python, PHP, …)

Other platforms (Android, iOS…)

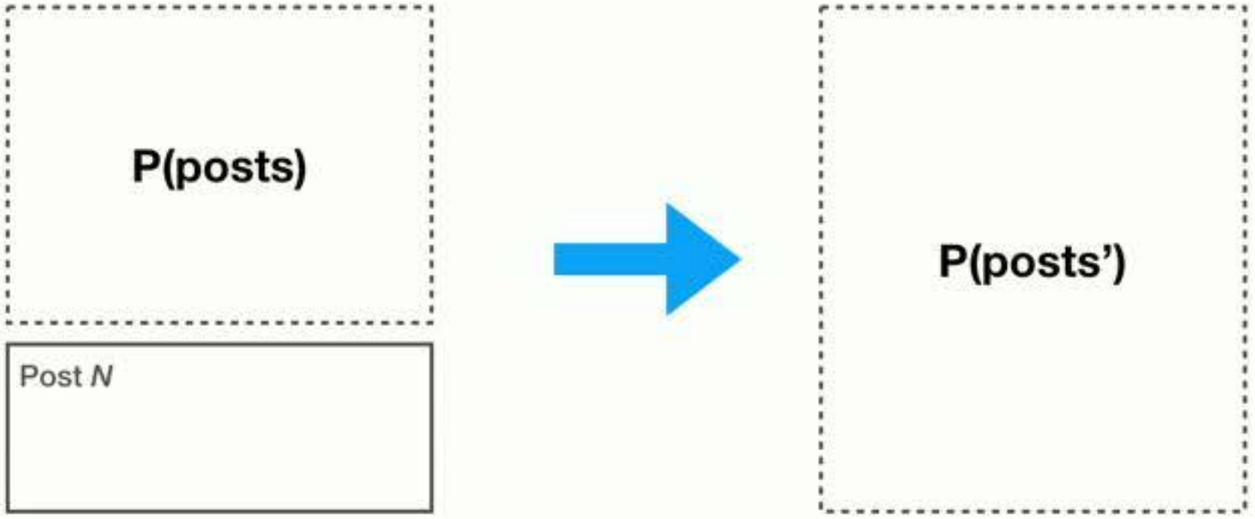
User studies with designers

Dynamic Web Pages



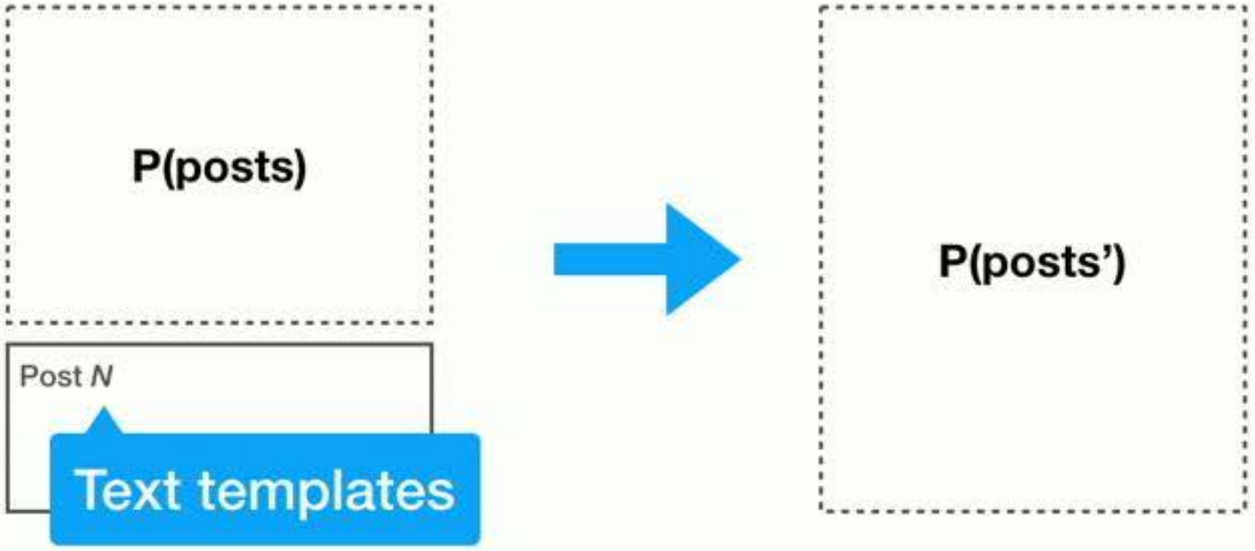
How can we verify pages if we don't know their contents?

Dynamic Web Pages



Induction over structure of the page

Dynamic Web Pages



Induction over structure of the page

Future Work

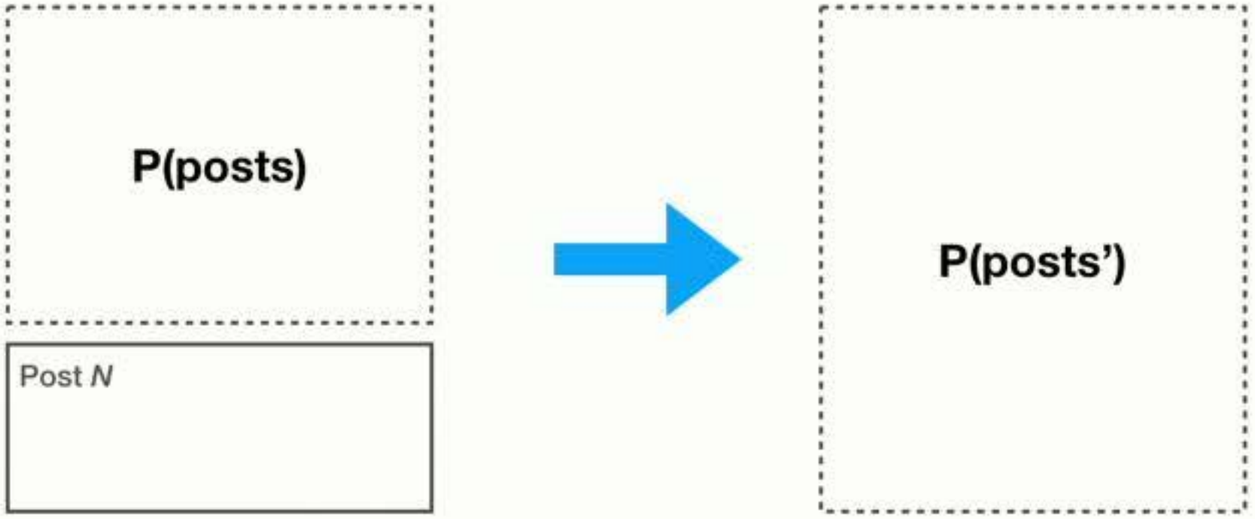
Frontend code (JavaScript)

Backend code (Python, PHP, …)

Other platforms (Android, iOS…)

User studies with designers

Dynamic Web Pages



Induction over structure of the page

Future Work

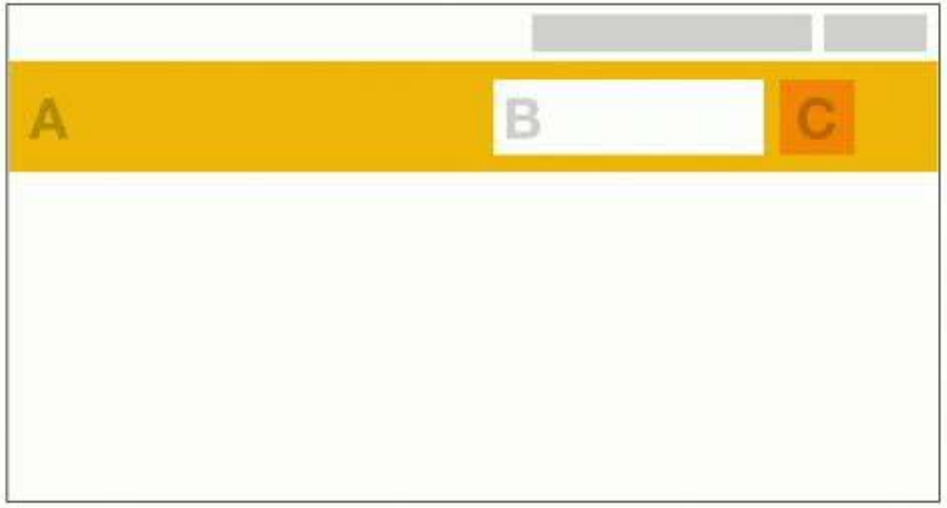
Frontend code (JavaScript)

Backend code (Python, PHP, …)

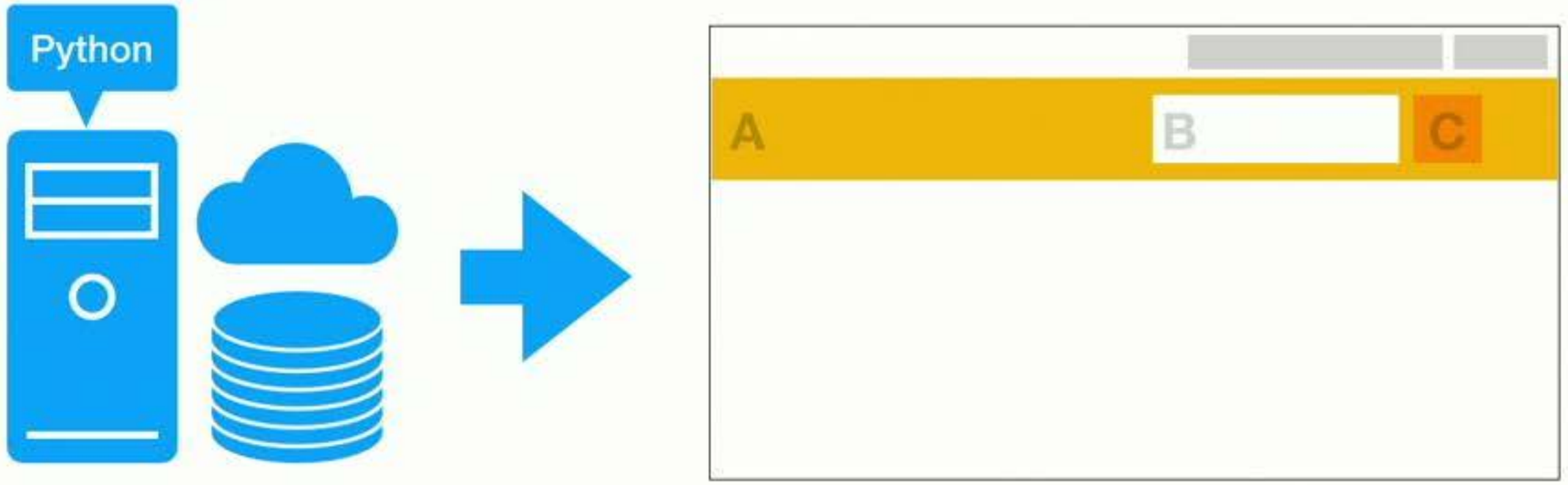
Other platforms (Android, iOS…)

User studies with designers

Web Page Backends



Web Page Backends



Web Page Backends

```
<div id='header'><h1>A</h1><form>…
```

```
<li class='search-result'>
  <h2>
    <a href="{obj.url}">{obj.name}</a>
  </h2>
  <p>{obj.description}</p>
</li>
```

Web Page Backends

```
write("<div id='header'><h1>A</h1><form>…")

for obj in results:
    write("<li class='search-result'>
             <h2>
               <a href="{obj.url}">{obj.name}</a>
             </h2>
             <p>{obj.description}</p>
           </li>")
```

Web Page Backends

for obj in results:

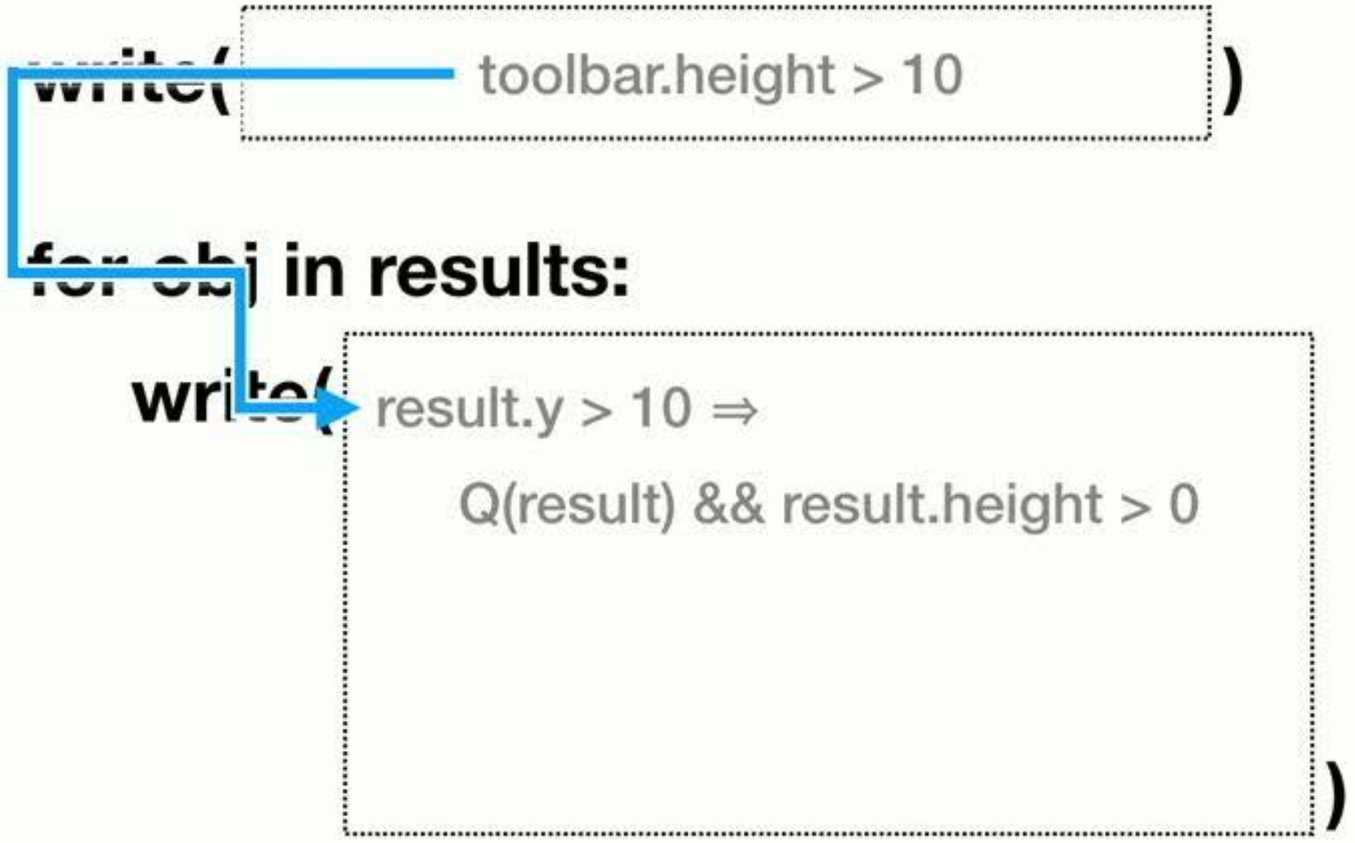
Web Page Backends

```
write( toolbar.height > 10 )

for obj in results:
    write( $obj.name
           $obj.description )
```

Web Page Backends

```
write( toolbar.height > 10 )

for obj in results:
    write( result.y > 10 ⇒
               Q(result) && result.height > 0
         )
```

Web Page Backends



Future Work

Frontend code (JavaScript)

Backend code (Python, PHP, …)

Other platforms (Android, iOS…)

User studies with designers

Future Work

Frontend code (JavaScript)

Backend code (Python, PHP, …)

Other platforms (Android, iOS…)

User studies with designers

Future Work

Frontend code (JavaScript)

Backend code (Python, PHP, …)

Other platforms (Android, iOS…)

User studies with designers

University of Utah Computer Science

University of Utah Computer Science

Exciting city in the middle of beautiful nature

Large & growing PL, compilers group

J. Regehr

Z. Rakamaric

G. Gopalakrishnan

P. Panchekha

P. Sadayappan

M. Hall

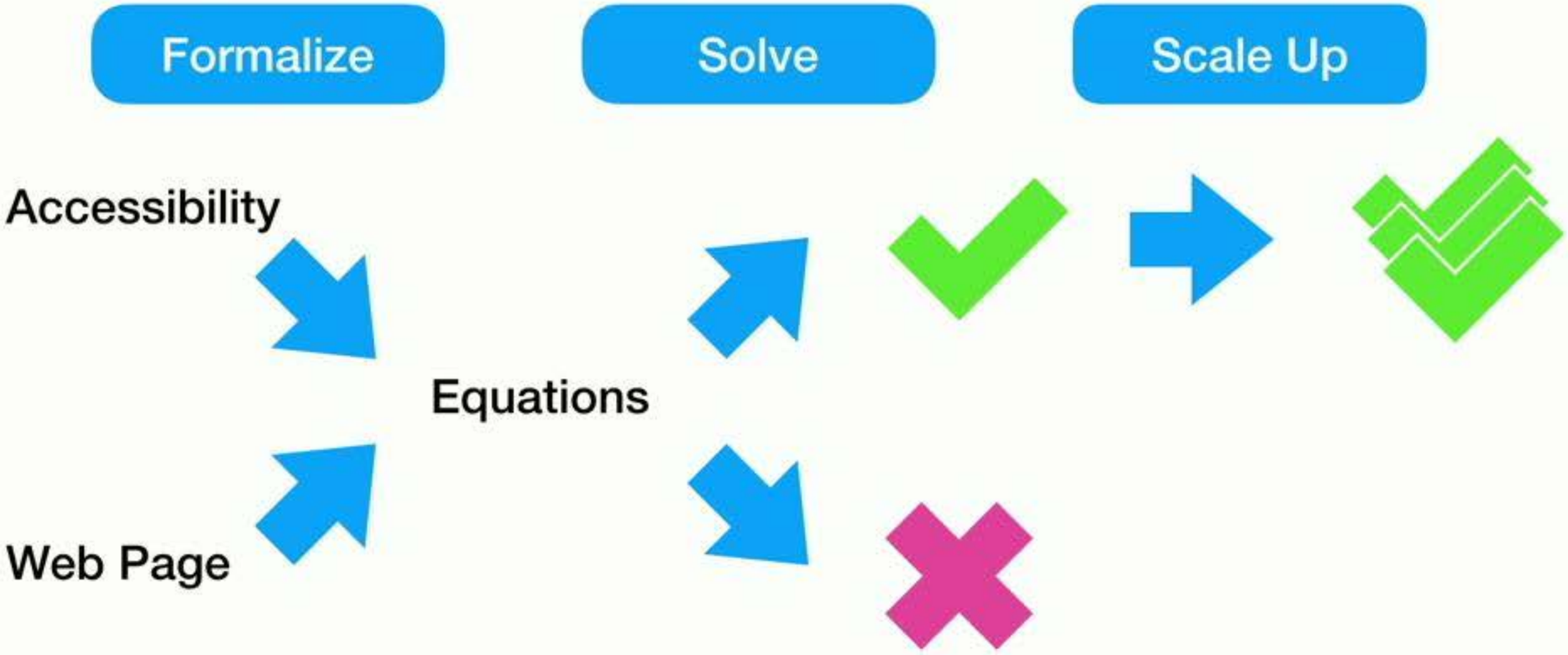
M. Flatt

University of Utah Computer Science

Exciting city in the middle of beautiful nature

Large & growing PL, compilers group

Thank You!

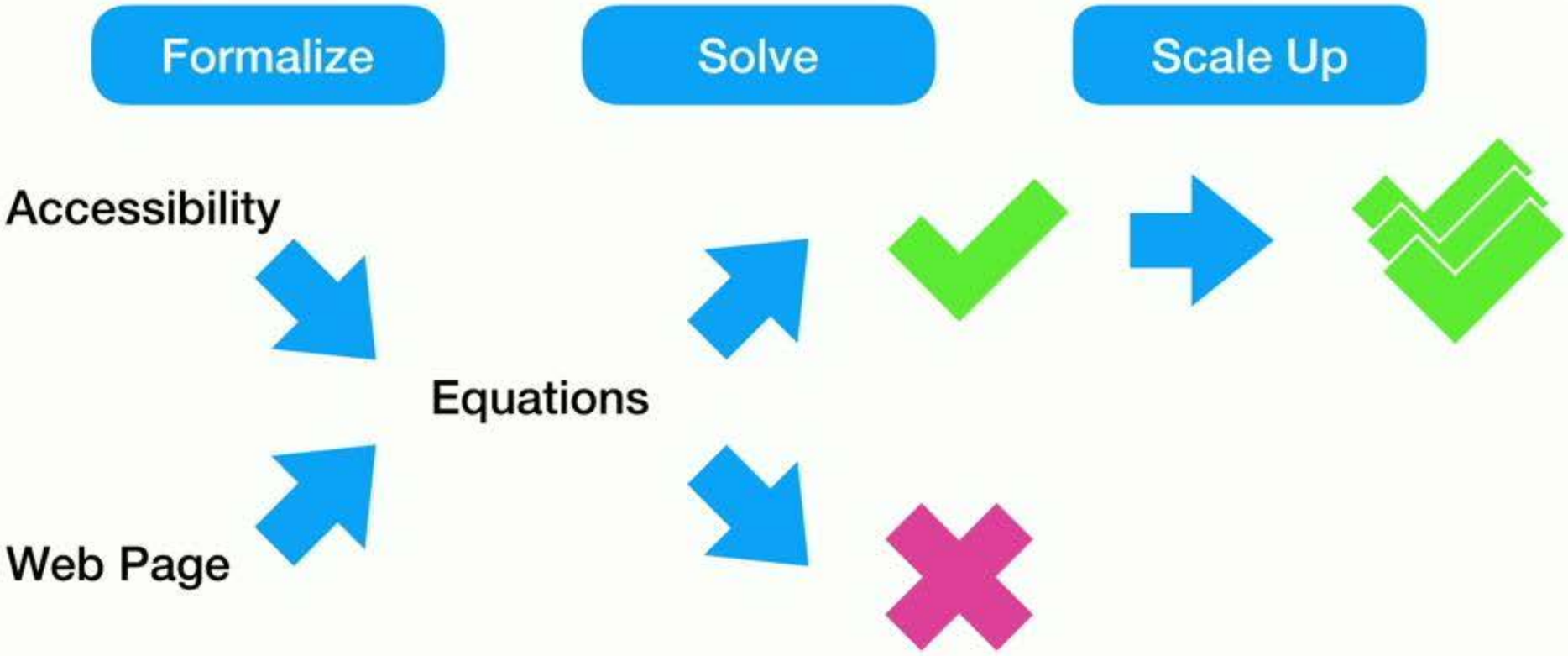


Formalizing visual properties

Overcoming solver limitations

Finding modularity for rely/guarantee

cassius.uwplse.org



Formalizing visual properties

Overcoming solver limitations

Finding modularity for rely/guarantee