Is HTML a context-free language?

Asked 13 years, 9 months ago Modified 11 years, 7 months ago Viewed 19k times

51

Reading some related questions made me think about the theoretical nature of HTML.

I'm not talking about XHTML-like code here. I'm talking about stuff like this crazy piece of markup, which is perfectly valid HTML(!)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html<head>
<title//
<p ltr<span id=p></span</p>
</>
```

So given the enormous complexity that SGML injects here, is HTML a context-free language? Is it a formal language anyway? With a grammar?

What about HTML5?

I'm new to the concept of formal languages, so please bear with me. And yes, I have read the wikipedia article ;)

html grammar language-theory sgml

Share
Improve this question
Follow

edited May 23, 2017 at 12:02
Community Bot
1 • 1

asked Mar 3, 2011 at 2:05
user123444555621
153k • 27 • 117 • 126

4 HTML is not context-free, as valid HTML code requires conditions which a CFG specification cannot handle (like unique `id` attributes etc..) – Nikos M. Jun 8, 2015 at 18:17

4 Answers

Sorted by: Highest score (default)

61

Context Free is a concept from language theory that has important implications in parser implementation. A *Context Free Language* can be described by a *Context Free Grammar*, which is one in which all rules have a single non-terminal symbol at the left of the arrow:

$X \rightarrow \delta$

That simple restriction allows `X` to be substituted by the right-hand side of the rules in which appears on the left without regard to what came before or after. For example, if while deriving or parsing one arrives at:

```
αXλ
```

one is sure that

```
αδλ
```

is also valid. Examples of non-context-free rules would be:

```
XY→δ
Xa→δ
aX→δ
```

Those would require knowing what could be derive arround `X` to determine if a rule applies, and that leads to non-determinism (what's around `X` would also like to know what it derives to), which is a no-no in parsing, and in any case we want a language to be well-defined.

The only way to prove that a language is context-free is by proving that there's a context-free grammar for it, which is not an easy task. Most programming languages one comes about are already described by CFGs, so the job is done. But there are other languages, including programming languages, that are described using logic or plain English, so work is required to find if they are context-free.

For HTML, the answer about its context-freedom is yes. SGML is a well defined Context Free Language, and HTML defined on top of it is also a CFL. Parsers and grammars for both languages abound on the Web. At any rate, that there exist LL(k) grammars for *valid* HTML is enough proof that the language is context-free, because LL is a proven subset of CF.

But the way HTML evolved over the life of the Web forced browsers to treat it as not that well defined. Modern Web browsers will go out of their way to try to render something sensible out of almost anything they find. The grammars they use are not CFGs, and the parsers are far more complex than the ones required for SGML/HTML.

HTML is defined at several levels.

1. At the lexical level there are the rules for valid characters, identifiers, strings, and so on.

2. At the next level is XML, which consists of the opening and closing `<tags>` that define a hierarchical document structure. You can use XML or something XML-like for any purpose, like `Apache Ant` does for build scripts.
3. At the next level are the tags that are valid in HTML, and the rules about which tags may be nested within which tags.
4. At the next level are the rules about which attributes are valid for which tags, languages that can be embedded in HTML like CSS and JavaScript.
5. Finally, you have the semantic rules about what a given HTML document means.

The syntactic part is defined well enough that it can be verified. The semantic part is much larger than the syntactic one, and is defined in terms of browser actions regarding HTTP, and the Document Object Model (DOM), and how a model should be rendered to the screen.

In the end:

1. Parsing correct HTML is extremely easy (it's context-free and LL/LR).
2. Parsing the HTML that actually exists over the Web is difficult.
3. Implementing the semantics (a browser) over HTML/CSS/DOM is extremely difficult.

Share

Improve this answer

Follow

edited Apr 27, 2013 at 7:13

answered Mar 6, 2011 at 0:02

Apalala

9,214 ● 3 ● 33 ● 50

Pretty much yes. So the reasoning for SGML being a CFL goes something like: "The specs ensure a well-defined parse tree. q.e.d." Meaning that the specs can be seen as some kind of informal CFG, right? (Example for browsers not following the rules: the <p><table> issue) – user123444555621 Mar 6, 2011 at 9:03

1 A well defined parse tree is the product of an unambiguous grammar, be it a CFG or not. Unambiguous means that if a sequence is part of the language, then there exists one an only one parse tree for it. See the edits for the "rules" part. – Apalala Mar 6, 2011 at 16:43

3 I am not sure #2 in your list is quite right. Valid XHTML is always valid XML, but valid HTML might not be valid XML. Did you mean SGML/DTD? One huge difference is the ability to omit/imply end tags in valid HTML (which SGML affords), but the inability to omit/imply end tags in XML. – Merlyn Morgan-Graham Oct 27, 2012 at 0:08

2 @Apalala Looking at the linked ANTLR file suggests that HTML's context-free status depends on there being a finite set of valid HTML tags (`<a>` , `<ul>` , etc). Is that correct? Am I right in thinking that arbitrary XML (eg `<foo></foo>` is context-sensitive? The name of the closing tag needs to match the name of the opening tag and you don't know what that name is. – Benjamin Hodgson Mar 27, 2017 at 13:21

2 @Benjamin Indeed. Formally, if the tags are not known in advance, then there cannot be a strict context-free grammar for XML. Yet, if we assume that the matching closing tags are on

the semantics level, then XML can be parsed by a CFG. Note that a language like Pascal can be parsed with an LL(1) grammar that doesn't try to check that identifiers are pre-declared and used with operators appropriate to their type. Most parsers used in practice do some semantic checks while parsing, as to detect obvious errors as early in the process as possible. – Apalala Mar 27, 2017 at 18:25

15

Valid HTML is not a context-free language.

First of all, HTML being an application of SGML is fiction for all practical purposes, so analyzing SGML to answer the question is useless. (However, the SGML fiction probably isn't context-free, either.)

It's more useful to look at the actually defined HTML parsing algorithm. It works on two levels: tokenization and tree building. What HTML calls tokenization is a higher-level operation than what is usually called tokenization when talking about parsers. In the case of HTML, tokenization splits a stream of characters into units like start tags, end tags, comments and text. The tokenizer expands character references. Usually, when talking about parsers, you'd probably treat stuff like the less-than sign as "tokens" and would consider character references to consist of tokens instead of being resolved by the tokenizer.

If you consider the process of splitting the input stream into tokens, that level of the HTML language is regular (*except* for feedback from the tree builder).

However, there are three complications: The first one is that splitting the input stream into tokens is just the first and then there's the tree builder's side that actually cares about the identifiers in the tokens. The second one is that the tree builder feeds back into the tokenizer so that some state transitions made by the tokenizer depend on the state of the tree builder! The third one is that valid documents in the language are defined by rules that apply to the output of the tree builder stage and those rules are complex enough that they can't be fully defined using tree automata (as evidenced by RELAX NG not being expressive enough to describe all the validity constraints).

This isn't an actual proof, but you can probably develop real proofs by working from complications #2 and #3.

Note that the case of invalid documents is not particularly interesting as a question of whether the language is context-free in the sense of there being a context-free grammar that generates all the possible strings with no regard to the parse tree having some intelligible interpretation in terms of the tree that an HTML parser generates. The HTML parser will successfully consume all possible strings, so in that sense, all possible strings are in the "invalid HTML" language.

Edit: Interesting questions left as exercise to the reader:

Is HTML without parse errors but ignoring validity a context-free language?

Is HTML without parse errors and ignoring general validity but with only valid element names allowed a context-free language?

(Complication #2 applies in both cases.)

Share

Improve this answer

Follow

edited Mar 6, 2013 at 11:17

answered Mar 6, 2013 at 10:27

hsivonen

8,006 ● 1 ● 35 ● 36

3 Which definition of *Context Free Language* are you using? – Apalala Mar 9, 2013 at 19:53

2 @Apalala, one simple example that HTML is not context-free is the fact that html `id` attributes must be unique, this cannot be described by any CFG,. but is part of HTML specification. see related question – Nikos M. Jun 8, 2015 at 15:22

9 @NikosM. Things like `id` should be handled by the semantic checker, not by the parser. If you went down that route basically all statically typed languages are not context-free because you need to type check the code. – semicolon Oct 3, 2016 at 18:18

1 @semicolon That's correct: basically all statically typed languages are context-sensitive. – mrr Feb 7, 2017 at 0:09

2 Well programs that don't type check are generally considered syntactically valid. The whitespace sensitive thing is a fair point, but I would still call a language context free even if you don't detect type errors in the process. – semicolon Feb 8, 2017 at 21:36

11

NO

See Edit Below

~~**It depends.**~~

~~If you are talking about the subset consisting of only theoretical HTML, then **yes**.~~

If you also include real life, working HTML that is accessed and used successfully by millions of people daily on many of the top sites on the internet then **NO**.

That is what gives HTML flexibility. The parsing engine adds tags, closes tags, and takes care of stuff that a theoretical CFG can't do. If you took automata you might remember that a production rule in a formal grammar cannot be empty (aka epsilon/lambda) on the lhs (left-hand side). Since the parsing engine is basically using knowledge that a formal grammar and automata couldn't have, it isn't restricted by that and the 'grammar' would have `epsilon/lambda -> result` where the specific epsilon/lambda rule is chosen based on information not available in the grammar.

Since I don't think empty lhs are allowed in any formal grammars, HTML cannot be defined by a formal grammar and is not a formal language at all.

Sure, HTML5 might try to move *towards* a 'more formal' language description but the likelihood that it becomes a context free language in reality (i.e. strings not matched by the grammar are rejected) is about the likelihood XHTML 2.0 takes the world by storm and replaces HTML altogether (XHTML is the attempt they made to make HTML a formal language...it was rejected en masse due to its fragility).

Noteworthy is the fact that HTML 5 is the FIRST HTML standard to be defined before being implemented! That's right, HTML 1-4 consist of random ideas someone just implemented in a browser, and were collected into standards after the fact based on which features were popularly used and widely implemented. Then they tried XHTML, which totally failed to be adopted. Even 'xhtml' on the web is automatically parsed as HTML under almost every circumstance to prevent stuff from just breaking with a cryptic syntax error. Now you can see how we got here and why it is unlikely to be formalized any time soon.

Lesson: "In theory, there is no difference between theory and practice. In practice, there is." - Yogi Berra

EDIT:

Actually, after reading through the documents it turns out that HTML, even according to the HTML 4.01 specification, doesn't actually conform to SGML. To see for yourself, view the HTML 4.01 Strict document type definition (doctype) at http://www.w3.org/TR/html4/strict.dtd and note the following lines:

> The HTML 4.01 specification includes additional syntactic constraints that cannot be expressed within the DTDs.

So I would say that it is *probably* not a CFL due to those features (although it technically it doesn't disprove the hypothesis that there is some possible PDA that accepts HTML 4.01, it does prevent the argument that SGML is a CFL therefore HTML is a CFL).

HTML5 flip-flops, abandoning any implied conformance to SGML, but is presumably describable by a CFG. However it will still provide best-effort parsing not based on a cfg, so IMO the current situation (i.e. language specification is defined formally, with invalid strings still being accepted, parsed and rendered in a best effort fashion) in this regard is unlikely to change drastically for a long, long, long time.

edited Nov 24, 2011 at 8:35

answered Nov 24, 2011 at 6:40

Follow

Brandon
1,970 ● 18 ● 19

I don't know which definition of *Context Free Language* you're using for your analysis. That DTDs cannot express all of HTML means nothing in regards to the original question. – Apalala Mar 9, 2013 at 19:52

2 @Apalala It has everything to do with the original question. Brandon is making the case (which you apparently disagree with) that HTML is not a subset of SGML, so it does not have the same grammar. More than that, his point is that the HTML "standard" (pre 5) is actually kind of an evolutionary mess, and so not really well defined. And DTDs are very much related to the formal definition of the language, making your comment difficult to understand. – shovavnik Sep 10, 2013 at 8:36

3 @shovavnik, Please see the most voted up answer in this query. In particular, that there are LL/LR parsers for "correct" HTML is enough proof that the language, as defined, is context free. That the evolution of the Web has required HTML parsers/renderers to dive into Artificial Intelligence to do their thing is awesome, but unrelated to the original question. – Apalala Sep 10, 2013 at 15:14

Saying that HTML isn't CFG because people can't write valid HTML is not a proof. I'm not saying that we don't need to take this into account. just that HTML is CFG. but we need to implement a lot of smart error handling if we need to implement a browser/interpreter – kam Dec 18, 2020 at 13:34

5

HTML5 is different from previous HTML versions in that it strictly defines the parsing behaviour of code that isn't completely correct. Pre-HTML5 parsers vary and each do their best to 'guess' the intention of the code author.

Share Improve this answer Follow

answered Mar 3, 2011 at 2:11

Delan Azabani
81.3k ● 30 ● 172 ● 212

2 Sure, but what does that mean in the context of grammar/language theory? – user123444555621 Mar 3, 2011 at 9:08

It means that in reality, it is not context free. – Brandon Nov 24, 2011 at 6:23

So you are saying that because people are bad at writing valid HTML, HTML are not CFG? – kam Dec 18, 2020 at 13:37