HTML Working Group D. Connolly
INTERNET-DRAFT MIT/W3C
draft-ietf-html-charset-harmful-00.txt May 2, 1995
Expires November, 1995

Character Set Considered Harmful

Status of this Document

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this document is unlimited. Please send comments to the HTML working group (HTML-WG) of the Internet Engineering Task Force (IETF) at `<html-wg@oclc.org>`;. Discussions of the group are archived at `http://www.acl.lanl.gov/HTML_WG/archives.html`.

Abstract

The term *character set* is often used to describe a ditigal representation of text. ASCII is perhaps the most widely deployed representation of text, and in the interest of interoperability, information systems on the Internet traditionally rely on it exclusively.

The Multipurpose Internet Mail Extensions (MIME) introduces Internet Media Types, including text representations besides ASCII. The Hypertext Markup Language (HTML) used in the World-Wide Web is a proposed Internet Media Type. But HTML is also an application of Standard Generalized Markup Language (SGML).

In the MIME and SGML specifications, the discussion of characters representation is notoriously complex, and apparently subtly inconsistent or incompatible. This document presents a collection of terms intended to reconcile the two specifications and serve as a basis for rigorous discussion of characters and their digital representations.

Introduction

The term *character set* is often used to describe a ditigal representation of text. The specification of such a representation typically involves identifying a sufficiently expressive collection of characters, and giving each of them a number.

In conventional mathematics terminology then, a "character set" is not just a set of characters, but a *function* whose domain is a set of integers, and whose range is a set of characters.

Some standards documents, including the SGML standard, make little or no use of such conventional mathematical terms as function, domain and range. Perhaps the authors of those documents intend the documents to be comprehensible without a prior understanding of mathematics. But the specification of notions such as the conformance of an SGML document or SGML system are much more complex than the basics of logic and mathematics.

In his text on Calculus[Spivak], Michael Spivak writes:

> Every aspect of this book was influenced by the desire to present calculus not merely as a prelude to but as the first real encounter with mathematics. Since the foundation of analysis provided the arena in which modern modes of mathematical thinking developed, calculus ought to be the place in which to expect, rather than avoid, the strengthening of insight with logic. In addition to developing the students' intuition about the beautiful concepts of analysis, it is surely equally important to persuade them that precision and rigor are neither deterrents to intuition, nor ends in themselves, but the natural medium in which to formulate and think about mathematical questions.

This document is not intended as the first real encounter with mathematics. But neither will we make any effort to avoid or apologize for mathematical terminology. The reader is referred to the large body of literature on logic and set theory, including a history of writings on math and logic[SET] and Douglas Hofstadter's fascinating book[GEB].

Coded Character Sets

Using "character set" rather than something such as *character table* or even *character sequence* to denote the functions that maps integers to characters is unfortunate, but it is water under the bridge, and a lot of it by now. Rather than attempting to divert all that water at this point, we introduce the primitive notion of character and use it to define the term *coded character set* from [ISO10646] and other standards:

character
: An atom of information

coded character set
: A function whose domain is a subset of the integers, and whose range is a set of characters.

Note that by the term character, we do not mean a glyph, a name, a phoneme, nor a bit combination. A character is simply an atomic unit of communication. It is typically a symbol whose various representations are understood to mean the same thing by a community of people.

It might seem more intuitive to map from characters to integers, rather than the way it is defined here. But in practice there are some coded character sets that assign two different numbers to the same character[Lee], and so the inverse is not a function in the general case.

There are two other terms used in standards such as [ISO10646] that we define in relation to the first two:

code position
: An integer. A coded character set and a code position from its domain determine a character.

character repertoire
: A set of characters; that is, the range of a coded character set.

Character Encoding Schemes

The only practical means for exchanging information on the Internet is to represent it as a sequence of octets (bytes).

One way to transmit a sequence of characters is to agree on a coded character set and transmit the character numbers of each of the characters.

But in practice, characters are encoded using a variety of optimizations of this brute-force approach: code switching techniques, escape sequences, etc. The encoding of a sequence of characters is not, in general, the result of encoding each character independently and then concatenating them. But it is sufficiently general to note that sequences of characters are encoded as a sequence of bytes. So we define:

octet
: an element of the set {0, 1, 2, ..., 255}

character encoding scheme
: a function whose domain is the set of sequences of octets, and whose range is the set of sequences of characters over some character repertoire.

Representation of SGML Text Entities

An SGML document is made up of entities: a text entity called the document entity, and possibly some other text entities and data entities.

A text entity is a sequence of characters. The representation of a text entity is not specified by the SGML standard. For the purpose of MIME-based interchange of SGML text entities, we define the following:

text entity
: a sequence of characters

message entity
: a pair (T, OS) where T is an Internet Media Type and OS is a sequence of octets.

Note that each `text/*` media type has an associated `charset` parameter, which designates a character encoding scheme. The character encoding scheme maps the body -- a sequence of octets -- to a text entity -- a sequence of characters. Hence any message entity of type `text/*` is equivalent to a text entity.

Numeric Character References

Numeric character references are a great source of confusion. The key insights are that:

- Every SGML document has exactly one document character set, which is a coded character set
- Numeric character references give code positions in the document character set

Example: ISO2022 Encoding with ISO10646 Coded Character Set

Consider the following message entity:

```
Date: Saturday, 29-Apr-95 03:53:33 GMT
MIME-version: 1.0
Content-Type: text/html; charset=iso-2022-jp

<TITLE>...</TITLE>
<BODY>
Here is some normal text.
Here is a 10646 numeric character reference  .
Here is some ISO-2022-JP text: ...
</BODY>
```

To interpret the message entity, we notice that the `Content-Type` is `text/html`, so this represents a text entity. The `charset` parameter `iso-2022-jp`, along with the octet sequence of the body, determines a sequence of characters. The octets denoted above by '...' represent characters, as per `iso-2022-jp`.

To parse the resulting text entity as per SGML, the sender and receiver must agree on an SGML declaration, since none is present in the document entity. For this example, we assume that SGML declaration specifies ISO10646 as the document character set. So the numeric character reference is resolved with respect to ISO10646.

It may seem contradictory that the `ISO-2022-JP` character encoding scheme is defined in terms of a collection of coded character sets, none of which is ISO10646. But there is no contradiction. Each character encoded by `ISO-2022-JP` is in the repertoire of one of those coded character sets, each of which is a subset of the repertoire of ISO10646.

So while `ISO-2022-JP` is not sufficient for every ISO10646 document, it is the case that ISO10646 is a sufficient document character set for any entity encoded with `ISO-2022-JP`.

Example: Reducing the Repertoire of an Entity

Suppose we have an SGML document D whose document character set is the coded character set ISO10646. We find the document entity DE in the form of sequence of octets OS in a disk file, encoded using the Unicode-UCS-2 character encoding scheme.

```
Unicode-UCS-2(OS) = DE
```

We can reduce the character repertoire necessary to represent the document entity by replacing characters outside the ISO-646-IRV character repertoire with numeric character references:

```
DE' = reduce(DE, ISO10646, ISO-646-IRV)
```

```
where

  reduce : SEQ(char) X Coded Character Set X Character Repertoire -> SEQ(char)

and

  reduce(c . rest, CCS, R) = if c in R, c . reduce(rest, CCS, R)
                                         else &#N; . reduce(rest, CCS, R)
                                         where CCS(N) = c
```

The resulting entity, DE' can then be endoded using US-ASCII

```
US-ASCII(OS') = DE' = reduce(DE, ISO10646, ISO-646-IRV)
```

Hence, we can represent the document D as a message entity whose content type is "text/plain; charset=US-ASCII" and whose body is OS'.

Conclusion

It is critical to keep separate the notion of a simple table of characters and their numbers, i.e. a coded character set, separate from the various algorithms to encoded sequences of characters, i.e. character encoding schemes. This separation allows a representation of a text entity which is consistent with both the MIME and SGML specifications.

Acknowledgements

The idea for the title of this document actually came from John Klensin. The notion of character encoding scheme was inspired by the MIME specification by Ned Freed. James Clark, Ed Levinson, and several other members of the MIMESGML working group collaborated in discussions leading up to this draft. Liam Quin from SoftQuad and Gavin Nicol from EBT have provided guidance on these issues in the past. Erik Naggum has provided invaluable aid in understanding the SGML standard.

References

[MIME]
N. Borenstein and N. Freed. "MIME (Multipurpose Internet Mail Extensions) Part One: Mechanisms for Specifying and Describing the Format of Internet Message Bodies." RFC 1521, Bellcore, Innosoft, September 1993.

[ASCII]
US-ASCII. Coded Character Set - 7-Bit American Standard Code for Information Interchange. Standard ANSI X3.4-1986, ANSI, 1986.

[ISO-8859]
ISO 8859. International Standard -- Information Processing -- 8-bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet No. 1, ISO 8859-1:1987. Part 2: Latin alphabet No. 2, ISO 8859-2, 1987. Part 3: Latin alphabet No. 3, ISO 8859-3, 1988. Part 4: Latin alphabet No. 4, ISO 8859-4, 1988. Part 5: Latin/Cyrillic alphabet, ISO 8859-5, 1988. Part 6: Latin/Arabic alphabet, ISO 8859-6, 1987. Part 7: Latin/Greek alphabet, ISO 8859-7, 1987. Part 8: Latin/Hebrew alphabet, ISO 8859-8, 1988. Part 9: Latin alphabet No. 5, ISO 8859-9, 1990.

[SGML]
ISO 8879. Information Processing -- Text and Office Systems -- Standard Generalized Markup Language (SGML), 1986.

[Nicol]
The Multilingual World Wide Web, Gavin T. Nicol, Electronic Book Technologies, Japan `gtn@ebt.com`

[Lee]
Private communication with Liam Quin, from SoftQuad.

[Spivak]
Spivak, Michael. Calculus. 2nd Ed. 1967 ISBN 0-914098-77-2

[GEB]
Hofstadter, Douglas R. Gödel, Escher, Bach: An Eternal Golden Braid, 1979 ISBN 0-394-75682-7

[SET]
"Investigations in the foundations of set theory I", in Jean van Heijenoort (ed.) _From Frege to Godel: A Source Book in Mathematical Logic, 1879-1931_ (Harvard U.P., 1967)