

satysfi-xml Manual

Naoki Kaneko a.k.a. "puripuri2100"

1. 概要

このライブラリは XML 文字列をパースしたり、逆にデータ構造を XML 文字列に変換したりする関数を提供するライブラリです。

W3C の公開している XML 1.0^{*1}が要求している要件のほぼ全てに対応していますが、External Entities という外部リソースを参照するための仕様には対応していません。

2. 提供するデータ構造

以下の型をトップレベルで定義しています。

テキストを表す型として `xml-text` という型を定義しています。 `XMLTextString` は素の文字列を、 `XMLTextEntityValue` は実体参照を表しています。よく使われる実体参照（角括弧やアンド記号、ユニコードスカラー値指定など）については参照されている値を取り出して `XMLTextString` に含めています。

```
1 type xml-text =  
2 | XMLTextString of string  
3 | XMLTextEntityValue of string
```

XML の要素を表す型として `xml-element` 型を定義しています。 `XMLTag` はタグの構成を表しています。中の値はそれぞれ

(1) タグ名

¹ <https://www.w3.org/TR/2008/REC-xml-20081126/>

- (2) 属性のリスト
- (3) 子要素のリスト

となっています。 `XMLText` はその名の通りテキストを表します。 `XMLProcessingInstructions` は `<? xml-stylesheet type="text/css"?>` のような処理命令を表します。中の値としてはそれぞれ

- (1) タグ名
- (2) 属性のリスト

を表します。

```
1 type xml-element =
2 | XMLTag of string * ((string * xml-text list) list) *
  ↪ (xml-element list)
3 | XMLText of xml-text
4 | XMLProcessingInstructions of string * ((string * xml
  ↪ -text list) list)
```

パースの結果で出てくる XML 構造体は `xml-element` のリストです。

```
1 type xml = xml-element list
```

解析に失敗した場合、失敗した理由を返します。それぞれの意味は

- `XMLParserErrorUnexpectedChar` : 来るはずのない文字が来た場合
- `XMLParserErrorUnexpectedString` : 文字列リテラル中に来るはずのない文字が来た場合
- `XMLParserErrorEOI` : 予期せぬ文字列の終了があった場合

です。

```
1 type xml-parser-error =
2 | XMLParserErrorUnexpectedChar of int * string
3 | XMLParserErrorUnexpectedString of int * int * string
```

4 | XMLParserErrorE0I

3. 提供する関数

モジュールは `XML` です。提供する関数は

- `parser : string -> xml xml-parser-error result`
- `printer : xml -> string`

の2つです。 `'ok 'err result` 型は `satysfi-base` ライブラリ内の `base.satyg` ファイル²内に定義されており、便利に操作するための関数は `result.satyg` ファイル³で提供されています。

4. 例

```
1 <hoge>
2   <?xml-stylesheet <!-- c -->type="text/css" href="tes ↵
   ↵ t.css"?>
3   fugahoge
4   <hoge2 test="hhh" test2="4&&test;2"/>
5   fuga<![CDATA[<sni>]]>ho<!--c-->ge
6 </hoge>
```

という XML コードを `XML.parser` という関数でパースすると

```
1 Ok(XMLTag(`hoge`, [], [
2   XMLProcessingInstructions(`xml-stylesheet`, [
```

2 <https://github.com/nyuichi/satysfi-base/blob/master/src/result.satyg>

3 <https://github.com/nyuichi/satysfi-base/blob/master/src/result.satyg>

```
3      (`type`, [XMLTextString(`text/css`)]);
4      (`href`, [XMLTextString(`tst.css`)]);
5  });
6  XMLText(`
7  fugahoge
8  `);
9  XMLTag(`hoge2`, [
10     (`test`, [XMLTextString(`hhh`)]);
11     (`test2`, [
12         XMLTextString(`4&`);
13         XMLTextEntityValue(`test`);
14         XMLTextString(`2`);
15     ]]);
16  ]);
17  XMLText(`
18  fuga<sni>hoge
19  `);
20  ]))
```

が返ってきます。