HTML 结构解析与校验实验报告

王松宸 2024201594

2025 年 11 月 3 日

1 核心内容

本实验基于线性表与栈，完成对 HTML 文档的读取、合法性校验（CheckHTML），并实现了 OuterHTML 与 Text 两个查询功能。

此外，**扩展实现了额外功能（a），支持输入绝对路径、相对路径与部分路径。**

同时，**针对用户实现了大小写不敏感匹配等功能**，提升了查询的灵活性与实用性。

2 代码总体设计

程序采用“单遍扫描 + 栈”策略：

1. 读取 HTML 文件（去除 UTF-8 BOM），缓存为 `g_html`；
2. 预处理阶段进行基础配对校验：遇到起始标签入栈，遇到结束标签出栈；完整跳过注释以及 `<script>`/`<style>` 块；将校验错误记录在 `g_errors` 中；
3. 交互命令：
   - CheckHTML：输出配对、嵌套、未闭合等错误；
   - Outer_HTML(path)：按简化 XPath 匹配，输出对应元素的 OuterHTML；
   - Text(path)：在匹配元素内抽取文本，按块级元素插入换行并合并多余空白。

核心模块关系如下：文件读取 → 预处理校验 → 路径解析与匹配 → OuterHTML/Text 输出。

3 数据结构与关键函数解析

3.1 栈结构

顺序栈 `SqStack`：

- 成员：`base`、`top`、`stacksize`；
- `initStack` 初始化容量为 `MAXSIZE`；`Push` 满则按 `MAXSIZE` 增量扩容；`Pop` 空栈返回错误；
- 额外提供 `PushTag`/`PopTag` 封装标签名的入栈/出栈（以 `'\0'` 作为分隔）。

3.2 路径解析与匹配（实现额外功能）

- **parse_path_tokens**：按 '/' 切分，去除多余空白；识别三类情形：
  1. 全部选择：路径为空或为 '/'，表示选择整个文档；
  2. 绝对路径：以 '/' 开头，要求路径与当前打开路径**完全相等**；
  3. 相对/部分路径：不以 '/' 开头，允许路径作为当前打开路径的**后缀**匹配（即部分路径）。
- **path_matches**：实现上述两种匹配策略；
- **perform_query**：扫描文档维护打开标签栈（名称序列），在遇到起始/结束标签时检查是否与目标路径匹配：
  - Outer_HTML：在匹配的起始标签处，找到与之配对的结束位置，输出完整片段；
  - Text：在匹配范围内调用文本抽取函数，输出归一化文本。

3.3 标签解析

- **find_tag_gt**：从字符 '<' 起，正确跨越属性单引号/双引号，找到配对的 '>'；
- **skip_comment**：跳过 <!-- ... --> 整段；
- **skip_script_style_block**：匹配对应的关闭标签，整体跳过脚本/样式块（不参与栈与内容抽取）；
- **自闭合识别**:内置常见自闭合标签集合（BR/HR/IMG/META/LINK/INPUT/AREA/...），并支持显式形式 <tag ... />；
- **大小写处理**：统一将标签名提升为大写参与比较，匹配不区分大小写。

3.4 文本抽取与归一化

- **extract_text_in_range**：忽略标签/注释/脚本样式，累积文本；遇到块级元素（HTML/BODY/DIV/P/UL/LI/TABLE/TD/TH/…）闭合时插入换行；
 直接视为换行；
- **normalize_text_preserve_newlines**：
  - 连续空格、\t、\r 折叠为单个空格；
  - 保留并折叠我们插入的换行符，逐行修剪首尾空格；
  - 移除末尾多余换行，确保输出整洁稳定。

3.5 预处理与内容校验

extbfpreprocess_html 负责：

- 线性扫描输入，遇起始标签入栈、结束标签出栈，记录未闭合/多余关闭等结构性错误；

- 对注释与 <script>/<style> 采取整体跳过，避免误报配对；
- 以 OpenTag 向量辅助报错位置输出。

extbfvalidate_content_model 对内容模型做检查，示例规则：

- R1：若当前为块级元素，祖先若为行内且不为 A，提示可能的嵌套不当；
- R2：祖先为 H1-H6/P/DT 时，不允许再出现块级后代；
- R3：A 不能包含 A 后代。

3.6 文件读入与 BOM 处理

extbfload_file 以二进制读取本地文件，统一移除 UTF-8 BOM，将内容缓存到全局 g_html 并以 '\0' 结尾，便于后续以 string_view 只读视图进行高效扫描。

4 命令接口说明

1 解析新的文件 读取本地 HTML 文件到内存，自动去除 UTF-8 BOM。

2 检查此文件合法性 利用栈完成配对、嵌套与未闭合检查，列出错误位置与标签名。

3 输出对应路径下的 html 代码段 Outer_html(path)：多个匹配节点用换行分隔。

4 输出对应路径下的文本 Text(path)：按块级元素换行策略输出整洁文本。

5 退出程序 结束交互。

5 面向用户的设计与交互细节

1 宽容的路径输入：路径前后空白会被修剪，多余的 '/' 自动忽略；标签名大小写不敏感；空路径或 '/' 表示“整页”。

2 清晰的提示与回显：未加载文件时阻止查询并提示操作顺序；每次命令后重新打印菜单，降低误操作成本。

3 输出一致性：多个匹配结果以单个换行分隔；文本抽取遵循“块级换行、空白合并、行级修剪”的格式化规则，便于直读与比对。

4 错误处理：

- 文件不存在/无法打开时及时报错；
- 结构性错误（未闭合、错位闭合、脚本/样式缺关闭等）集中展示；
- 查询无命中时输出提示而非静默失败。

6 总结

本实验以顺序栈为核心，完成了 HTML 的结构校验与两类查询；在此基础上实现了 XPath 额外功能（绝对/相对/部分路径）与用户友好功能。