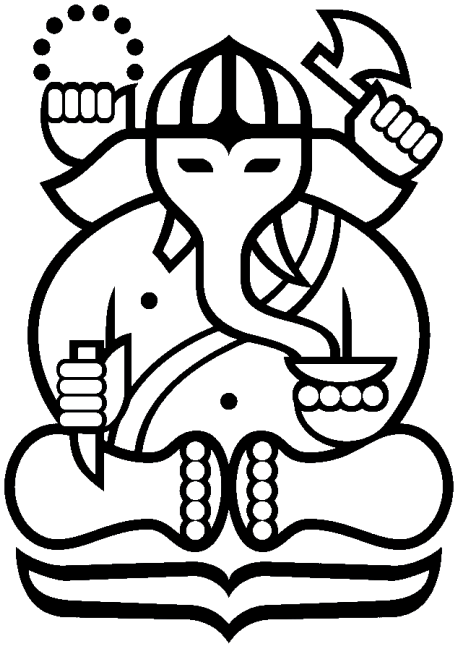
LAPORAN TUGAS BESAR

TEORI BAHASA FORMAL DAN OTOMATA

Dosen: Judhi Santoso



Kelompok 15

Jimmy 88

Daftar Anggota

1. Zachary Samuel T / 13522016
2. Wilson Yusda / 13522019
3. Benardo / 13522055

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

2023

Daftar Isi

Daftar Isi **2**

Bab I
Teori Dasar **4**

1.1. HTML 4
1.2. Pushdown Automata 6
1.3. Pengaplikasian PDA dalam HTML 7

Bab II
Hasil Pushdown Automata **9**

1.1. Hasil PDA 9
1.2. Finite Set of States (Q) 10
1.3. Alphabet (Σ) 11
1.4. Stack Alphabet (Γ) 11
1.5. Transition Function (δ) 11
1.6. Start State (q0) 40
1.7. Stack Symbol (Z0) 40
1.8. Accepting States (F) 41

Bab III
Implementasi dan Pengujian **42**

1.1. Spesifikasi Teknis Program 42
1.1.1. Struktur Data 42
1.1.2. Fungsi dan Prosedur 42
1.1.3. Antarmuka 46
1.2. Uji Kasus 46
1.2.1. Kasus 1 46
1.2.2. Kasus 2 47
1.2.3. Kasus 3 48
1.2.4. Kasus 4 48
1.2.5. Kasus 5 49
1.2.6. Kasus 6 50
1.2.7. Kasus 7 50
1.2.8. Kasus 8 51
1.2.9. Kasus 9 52
1.2.10. Kasus 10 53
1.2.11. Kasus 11 54

Bab IV
Tautan dan Pembagian Tugas.....56
1.1. Link Repository GitHub..... 56
1.2. Link Diagram State.....56
1.3. Pembagian Tugas..... 56
Bab V
Daftar Pustaka.....57

Bab I
Teori Dasar

1.1. HTML

HTML (*HyperText Markup Language*) adalah bahasa standar yang digunakan untuk membuat dan mengatur struktur halaman web. HTML digunakan untuk mengatur struktur halaman web dan kontennya. Ini bisa mencakup pengorganisasian konten dalam paragraf, daftar (berpoin atau bernomor), atau menggunakan gambar dan tabel data. Dalam HTML, tersinggung beberapa jenis tag atau kelompok dengan fungsi yang berbeda.

Tag ini menjadi ciri khas HTML yang digunakan sebagai *syntax* dari bahasanya yang digunakan untuk mendefinisikan dan mengklasifikasikan bagian-bagian dari *web* dengan berbagai fungsi lain seperti *container, styling,* fungsi tambahan (*hyperlink*), dan berbagai struktur (tabel, *list*).

Beberapa diantaranya yaitu:

1. html, merupakan sebuah tag yang wajib ada. Merupakan sebuah tag yang selalu berada di posisi paling luar dibandingkan dengan tag lainnya
2. head, merupakan sebuah tag *container* yang berfungsi untuk menampung tag tag tertentu seperti <title>, <style>, <meta>, <link>, <script>, dan <base>
3. body, merupakan sebuah tag yang wajib ada dalam sebuah file HTML berfungsi untuk menampung sebagian besar tag dalam HTML
4. title, mendefinisikan judul sebuah dokumen
5. link, mengarahkan suatu file html dengan file luaran
6. script, mengaplikasikan suatu *script* pada sebuah laman (umumnya JavaScript)
7. h1, h2, h3, h4, h5, h6, menuliskan *headings* pada laman

8. p, menuliskan sebuah tampilan *text* pada laman
9. br, memisahkan sebuah *text* menjadi 2 posisi secara vertikal
10. em, memberikan penekanan pada sebuah *text*
11. b, memberikan sebuah *text font* tebal
12. abbr, mendefinisikan sebuah *text* sesuai singkatannya
13. strong, mendefinisikan *text* dengan kepentingan kuat, biasanya juga dalam *bold*
14. small, mendefinisikan text secara kecil, biasanya seperti *copyrights*
15. hr, tag yang berfungsi membagi bagian pada laman
16. div, merupakan elemen yang berfungsi layaknya kotak dengan ukuran tertentu
17. a, mengarahkan pengguna ke laman luar
18. img, menampilkan foto pada laman
19. button, menampilkan tombol yang bisa dihubungkan ke fitur
20. form, menarik inputan dari pengguna
21. input, umumnya diaplikasikan pada *form*, menerima *input* dengan berbagai cara
22. table, menampilkan tabel pada laman
23. tr, mendefinisikan baris pada tabel
24. td, mendefinisikan satuan data *cell* standard
25. th, mendefinisikan *header cell* pada HTML

Dalam HTML, istilah "nested elements" dan "void elements" merujuk pada dua konsep yang berbeda dalam struktur markup halaman web. “Nested" dalam konteks HTML berarti elemen yang ditempatkan di dalam elemen lain. Ini sering digunakan untuk mendefinisikan struktur dan hierarki konten pada halaman web. Void elements, atau elemen kosong, adalah elemen HTML yang tidak memerlukan tag penutup karena mereka tidak dapat memuat konten. Ini berbeda dari elemen lain yang biasanya memiliki tag pembuka dan penutup.

Dalam HTML, ada juga tipe bernama atribut yang memberikan informasi tambahan tentang elemen HTML dan membantu mendefinisikan atau mengatur cara elemen tersebut akan ditampilkan atau berinteraksi dengan halaman web. Atribut ditempatkan dalam tag HTML dan selalu berada di dalam tanda kurung (< >). Atribut terdiri juga dari atribut global yang dapat berada dalam semua tag seperti *class*, *id*, *style*, dan atribut yang spesifik pada setiap tag seperti img yang terdiri dari src (*source*) dan alt (teks alternatif).

1.2. Pushdown Automata

Pushdown Automaton (PDA) merupakan konsep penting dalam teori komputasi, yang merupakan cabang dari ilmu komputer teoritis. PDA adalah jenis automaton yang menggunakan stack. Stack ini memungkinkan PDA untuk menyimpan simbol sementara dan melakukan operasi push (menambahkan simbol di atas stack) dan pop (mengeluarkan simbol dari stack) selama transisi antar keadaan.

Komponen-komponen pada PDA yakni :

- Q: Kumpulan keadaan terbatas.
- ∑: Kumpulan simbol input.
- Γ: Kumpulan simbol pushdown yang dapat didorong dan dikeluarkan dari stack.
- q0: Keadaan awal.
- Z: Simbol pushdown awal yang awalnya ada di dalam stack.
- F: Kumpulan keadaan akhir

Komponen-komponen PDA ini yang kemudian digunakan sebagai simbol penulisan sebuah PDA. Dalam tugas ini, digunakan *format* penulisan *current state* (salah satu state dari Q), *input* yang dibaca (salah satu dari ∑ dengan pengecualian e sebagai *epsilon transition*), simbol yang diambil dari stack (salah satu dari Γ), *next state* (salah satu dari Q), dan simbol yang ditambahkan pada stack (salah satu

dari Γ dengan penulisan yang baru antara *push* pada stack, mengganti simbol *top stack*, dan *pop* stack).

Penerapan Pushdown Automata (PDA) dalam pengecekan HTML mengacu pada penggunaan PDA untuk memvalidasi struktur dokumen HTML. Struktur berjenjang dan bersarang dari elemen HTML sangat mirip dengan bahasa kontekstual yang dapat diakomodasi oleh PDA.

Dalam konsep PDA juga diterapkan 2 jenis PDA, yaitu PDA dengan penerimaan *empty stack* (ketika *stack* kosong), dan *final state* (sudah mencapai *state* tertentu). Dalam pengecekan HTML, dilihat lebih sesuai bahwa PDA yang dibuat menerima dengan *empty stack* karena sifatnya yang memeriksa *syntax* sehingga dapat dilihat jumlah *opening* dan *closing* tag serta simbol-simbol lain agar sesuai dengan ketentuan penulisan HTML.

Dalam HTML, tag harus ditutup dalam urutan yang benar; misalnya, jika Anda memiliki tag <b> di dalam tag <p>, tag <b> harus ditutup sebelum tag <p> ditutup. Ini sangat mirip dengan konsep LIFO (*Last In, First Out*) dalam *stack*, yang merupakan komponen utama dari PDA. PDA dapat dimodelkan untuk menerima *string* HTML dan menggunakan stack untuk melacak tag-tag yang dibuka. Setiap kali sebuah tag pembuka ditemui, PDA akan *push* tag itu ke stack.

Ketika tag penutup ditemui, PDA akan *pop* tag dari stack dan memeriksa apakah tag tersebut cocok dengan tag penutup. Jika cocok, proses berlanjut; jika tidak, ini menunjukkan kesalahan dalam struktur HTML.

1.3. Pengaplikasian PDA dalam HTML

Pemanfaatan Pushdown Automata (PDA) dalam menganalisis dan memvalidasi struktur HTML menunjukkan aplikasi teori komputasi yang efisien dalam bidang pengembangan web. HTML, sebagai bahasa markup, memiliki kemiripan dengan bahasa konteks-bebas yang bisa diuraikan menggunakan

konsep-konsep yang ada dalam PDA. Struktur HTML yang cenderung memiliki elemen yang bersarang, di mana tag tertentu terletak di dalam tag lain, menciptakan hierarki serupa dengan yang terdapat dalam Context-Free Grammar (CFG). PDA memanfaatkan stack yang berbasis prinsip LIFO (Last In, First Out) untuk mengawasi tag-tag HTML yang telah dibuka dan belum ditutup.

Dalam proses parsing HTML dengan PDA, ada beberapa tahapan utama. Saat sebuah tag pembuka seperti ditemui, PDA akan menumpuknya ke dalam stack, menandakan permulaan sebuah elemen bersarang. Kemudian, ketika sebuah tag penutup seperti muncul, PDA akan mengeluarkan tag dari stack dan memverifikasi kecocokannya dengan tag penutup yang ditemukan. Kesesuaian ini menandakan kelengkapan blok HTML dan memungkinkan proses parsing dilanjutkan; ketidaksesuaian menunjukkan keberadaan kesalahan struktur dalam HTML yang harus diperbaiki.

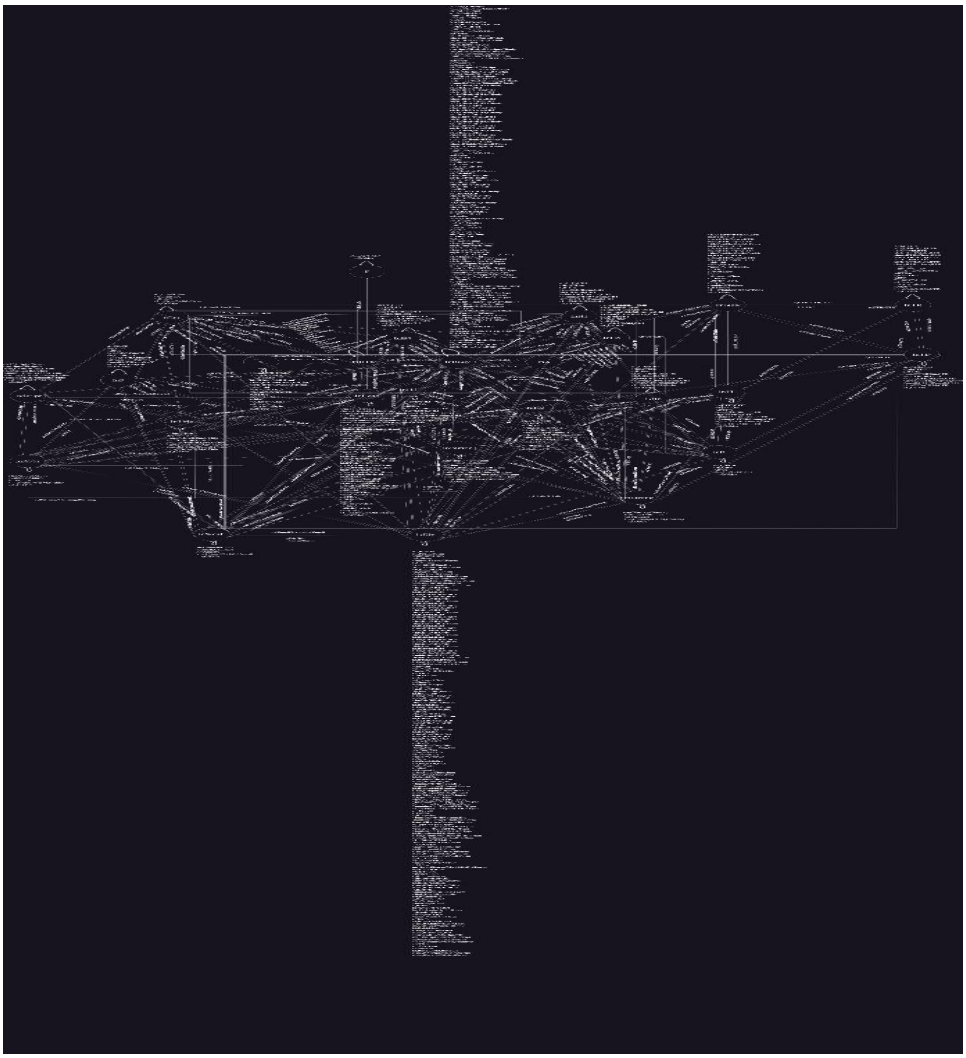
Selain itu, PDA menggunakan stack untuk validasi struktural HTML, memastikan setiap tag pembuka memiliki tag penutup yang cocok dan bahwa penempatan tag-tag tersebut sesuai. Hal ini krusial untuk mencegah kesalahan sintaks yang bisa mempengaruhi tampilan halaman web. PDA juga mengatur elemen-elemen kosong dan elemen bersarang di HTML. Untuk elemen-elemen yang tidak membutuhkan tag penutup (void elements), PDA hanya akan mengabaikannya dalam urusan pengelolaan stack. Sementara untuk elemen-elemen yang bersarang, PDA akan menanganinya secara akurat melalui operasi penumpukan dan pengeluaran untuk menjaga integritas struktur dokumen HTML. Pendekatan ini membuat PDA tidak hanya sebagai alat untuk validasi HTML, tetapi juga membantu memahami struktur dokumen HTML lebih dalam. Penerapan konsep PDA dalam konteks HTML menunjukkan bagaimana prinsip-prinsip teori komputasi diterapkan dalam pengembangan web praktis. Ini menjamin pembuatan dokumen HTML yang tepat, yang sangat penting untuk pengalaman pengguna yang optimal di internet. Dengan demikian, PDA menyediakan fondasi teoritis yang solid bagi siapa saja yang terlibat dalam pengembangan dan analisis HTML, meningkatkan pemahaman mereka tentang struktur dan standar kode yang efektif.

Bab II

Hasil Pushdown Automata

1.1. Hasil PDA

Berikut terlampir gambar PDA yang kami buat.



Untuk lebih jelasnya, dapat dilihat pada bagian Link Diagram State pada Bab 4 dengan mengakses link yang dilampirkan.

1.2. Finite Set of States (Q)

Dalam pengaplikasian tugas besar ini, beberapa state yang kami buat guna menerapkan PDA yakni:

Q =

1. S (Start)
2. InHTML(Berada di dalam tag HTML)
3. InTitle (Berada di dalam tag Title)
4. InBody (Berada di dalam tag Body)
5. InScript (Berada di dalam tag Script)
6. InH1 (Berada di dalam tag H1)
7. InH2 (Berada di dalam tag H2)
8. InH3 (Berada di dalam tag H3)
9. InH4 (Berada di dalam tag H4)
10. InH5 (Berada di dalam tag H5)
11. InH6 (Berada di dalam tag H6)
12. InP (Berada di dalam tag P)
13. inEm (Berada di dalam tag Em)
14. inAbbr (Berada di dalam tag Abbr)
15. inB (Berada di dalam tag B)
16. inSmall (Berada di dalam tag Small)
17. inStrong (Berada di dalam tag Strong)
18. InA (Berada di dalam tag a)
19. InTable (Berada di dalam tag table)
20. InTR (Berada di dalam tag TR)
21. InTD (Berada di dalam tag TD)
22. InTH (Berada di dalam tag TH)
23. InButton (Berada di dalam tag Button)
24. InDiv (Berada di dalam tag Div)
25. F (Final state)

1.3. Alphabet (Σ)

Dalam merancang PDA, terdapat kumpulan input symbol yang diperoleh dari parsing HTML.

Σ = {</small, </h6, id, <!-- ,"submit", <th, </h5, "POST", <hr, </h2, <br, </head, <h6, <title, <form ,<p ,</p, </form ,style ,"number" ,</h1, </body, alt, <abbr, <html, </b, <tr, <button ,class, <script, rel, </h3, </html, </abbr, <div, </table, src, >,method, <input, action ,"text", </div, <h5, "checkbox", </strong ,<b, -->, </a ,<small, <table ,"password" ,"nostr", <img ,<link, </tr, <head ,<h4 ,href ,"reset", <h3 ,e ,str ,</script ,"email" ,</button ,"button" ,<td, <em ,</td, </em ,<a ,<strong, "GET", =, <h1, </h4, </title, type ,</th, "str" ,<body ,<h2 }

1.4. Stack Alphabet (Γ)

Γ = {CH2, BEm , CH6, BTable, BHtml, id ,BH2 ,CA ,CB ,Z ,Src, NOSTR ,Href, BStrong ,CH4 ,COM ,CTD ,BP ,CAbbr, CTitle ,BH3 ,CH3 ,BBody ,CBody ,Type, CScript ,BH6 ,CH1 ,BTH ,BA ,CTable ,Div ,CEm ,CSmall, BScript ,Form, CStrong, X ,Class, BButton, BB ,BForm ,CButton, STR ,Rel, ACT ,Img ,BDiv, BInput, BR ,BH5 ,BTD ,CP ,Style, BAbbr, BSmall, BTR ,CDiv ,Source ,BH1 ,CForm ,BTitle ,=Rel ,CHead ,=Src ,METHOD ,BH4 ,CH5 ,= ,BHead ,type, =Type, Alt, CTH ,HR ,CInput ,BLink ,CHtml ,CTR ,=Source }

1.5. Transition Function (δ)

Untuk melakukan proses penerapan PDA pada tubes ini, kami membuat beberapa transition function yang dibutuhkan sebagai suatu aturan untuk memproses token dengan rancangan PDA . Berikut rancangan transition function yang telah dibuat.

a. Current state : S

δ(S,<html,Z) = {(S, BHtmlZ)}

δ(S,>,BHtml) = {(InHTML, CHtmlBHtml)}

δ(S,<!--,Z) = {(S, COMZ)}

δ(S,-->,COM) = {(S, e)}

δ(S,id,BHtml) = {(S, idBHtml)}

δ(S,class,BHtml) = {(S, ClassBHtml)}

δ(S,style,BHtml) = {(S, StyleBHtml)}

δ(S,=,id) = {(S, =id)}

δ(S,"str",=) = {(S, STR=)}

δ(S,"nostr",=) = {(S, NOSTR=)}

δ(S,e,STR) = {(S, e)}

δ(S,e,NOSTR) = {(S, e)}

δ(S,e,=) = {(, S)}

δ(S,e,id) = {(S, e)}

δ(S,=,Class) = {(S, =Class)}

δ(S,e,Class) = {(S, e)}

δ(S,=,Style) = {(S, =Style)}

δ(S,e,Style) = {(S, e)}

b. Current state : InHtml

δ(InHTML,<head,CHtml) = {(InHTML, BHeadCHtml)}

δ(InHTML,id,BHead) = {(InHTML, idBHead)}

δ(InHTML,class,BHead) = {(InHTML, ClassBHead)}

δ(InHTML,style,BHead) = {(InHTML, StyleBHead)}

δ(InHTML,>,BHead) = {(InHead, CHeadBHead)}

δ(InHTML,<body,X) = {(InHTML, BBody)}

δ(InHTML,>,BBody) = {(InBody, CBodyBBody)}

δ(InHTML,</html,CHtml) = {(InHTML, e)}

δ(InHTML,>,BHtml) = {(F, e)}

δ(InHTML,=,id) = {(InHTML, =id)}

δ(InHTML,"str",=) = {(InHTML, STR=)}

δ(InHTML,"nostr",=) = {(InHTML, NOSTR=)}

δ(InHTML,e,STR) = {(InHTML, e)}

δ(InHTML,e,NOSTR) = {(InHTML, e)}

δ(InHTML,e,=) = {(, InHTML)}

δ(InHTML,e,id) = {(InHTML, e)}

δ(InHTML,=,Class) = {(InHTML, =Class)}

δ(InHTML,e,Class) = {(InHTML, e)}

δ(InHTML,=,Style) = {(InHTML, =Style)}

δ(InHTML,e,Style) = {(InHTML, e)}

δ(InHTML,<!--,CHtml) = {(InHTML, COMCHtml)}

δ(InHTML,-->,COM) = {(InHTML, e)}

δ(InHTML,<!--,X) = {(InHTML, COMX)}

c. Current state : InHead

δ(InHead,</head,CHead) = {(InHead, e)}

δ(InHead,>,BHead) = {(InHTML, X)}

δ(InHead,<title,CHead) = {(InHead, BTitleCHead)}

δ(InHead,>,BTitle) = {(InTitle, CTitleBTitle)}

δ(InHead,<script,CHead) = {(InHead, BScriptCHead)}

δ(InHead,>,BScript) = {(InScript, CScriptBScript)}

δ(InHead,src,BScript) = {(InHead, SourceBScript)}

δ(InHead,=,Source) = {(InHead, =SourceSource)}

δ(InHead,"str",=Source) = {(InHead, STR=Source)}

δ(InHead,"nostr",=Source) = {(InHead, NOSTR=Source)}

δ(InHead,e,=Source) = {(InHead, e)}

δ(InHead,e,Source) = {(InHead, e)}

δ(InHead,<link,CHead) = {(InHead, BLinkCHead)}

δ(InHead,rel,BLink) = {(InHead, Rel)}

δ(InHead,=,Rel) = {(InHead, =RelRel)}

δ(InHead,"str",=Rel) = {(InHead, e)}

δ(InHead,"nostr",=Rel) = {(InHead, e)}

δ(InHead,>,Rel) = {(InHead, e)}

δ(InHead,href,Rel) = {(InHead, HrefRel)}

δ(InHead,href,BLink) = {(InHead, HrefBLink)}

δ(InHead,=,Href) = {(InHead, =Href)}

δ(InHead,"str",=) = {(InHead, STR=)}

δ(InHead,"nostr",=) = {(InHead, NOSTR=)}

δ(InHead,e,Href) = {(InHead, e)}

δ(InHead,id,BHead) = {(InHead, idBHead)}

δ(InHead,class,BHead) = {(InHead, ClassBHead)}

δ(InHead,style,BHead) = {(InHead, StyleBHead)}

δ(InHead,id,BTitle) = {(InHead, idBTitle)}

δ(InHead,class,BTitle) = {(InHead, ClassBTitle)}

δ(InHead,style,BTitle) = {(InHead, StyleBTitle)}

δ(InHead,id,BLink) = {(InHead, idBLink)}

δ(InHead,class,BLink) = {(InHead, ClassBLink)}

δ(InHead,style,BLink) = {(InHead, StyleBLink)}

δ(InHead,id,BScript) = {(InHead, idBScript)}

δ(InHead,class,BScript) = {(InHead, ClassBScript)}

δ(InHead,style,BScript) = {(InHead, StyleBScript)}

δ(InHead,=,id) = {(InHead, =id)}

δ(InHead,"str",=) = {(InHead, STR=)}

δ(InHead,"nostr",=) = {(InHead, NOSTR=)}

δ(InHead,e,STR) = {(InHead, e)}

δ(InHead,e,NOSTR) = {(InHead, e)}

δ(InHead,e,=) = {(, InHead)}

δ(InHead,e,id) = {(InHead, e)}

δ(InHead,=,Class) = {(InHead, =Class)}

δ(InHead,e,Class) = {(InHead, e)}

δ(InHead,=,Style) = {(InHead, =Style)}

δ(InHead,e,Style) = {(InHead, e)}

δ(InHead,<!--,CHead) = {(InHead, COMCHead)}

δ(InHead,-->,COM) = {(InHead, e)}

d. Current state : InTitle

δ(InTitle,</title,CTitle) = {(InTitle, e)}

δ(InTitle,>,BTitle) = {(InHead, e)}

δ(InTitle,str,CTitle) = {(InTitle, CTitle)}

δ(InTitle,<em,CTitle) = {(InTitle, BEmCTitle)}

δ(InTitle,>,BEm) = {(inEm, CEmBEm)}

δ(InTitle,<b,CTitle) = {(InTitle, BBCTitle)}

δ(InTitle,>,BB) = {(inB, CBBB)}

δ(InTitle,<abbr,CTitle) = {(InTitle, BAbbrCTitle)}

δ(InTitle,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InTitle,<strong,CTitle) = {(InTitle, BStrongCTitle)}

δ(InTitle,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InTitle,<small,CTitle) = {(InTitle, BSmallCTitle)}

δ(InTitle,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InTitle,<!--,CTitle) = {(InTitle, COMCTitle)}

δ(InTitle,-->,COM) = {(InTitle, e)}

e. Current state : InScript

δ(InScript,str,CScript) = {(InScript, CScript)}

δ(InScript,</script,CScript) = {(InScript, e)}

δ(InScript,>,BScript) = {(InScript, e)}

δ(InScript,e,CHead) = {(InHead, CHead)}

δ(InScript,e,CBody) = {(InBody, CBody)}

δ(InScript,e,Div) = {(InDiv, Div)}

δ(InScript,e,Form) = {(InDiv, Form)}

δ(InScript,<em,CTitle) = {(InScript, BEmCTitle)}

δ(InScript,>,BEm) = {(inEm, CEmBEm)}

δ(InScript,<b,CTitle) = {(InScript, BBCTitle)}

δ(InScript,>,BB) = {(inB, CBBB)}

δ(InScript,<abbr,CTitle) = {(InScript, BAbbrCTitle)}

δ(InScript,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InScript,<strong,CTitle) = {(InScript, BStrongCTitle)}

δ(InScript,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InScript,<small,CTitle) = {(InScript, BSmallCTitle)}

δ(InScript,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InScript,<!--,CScript) = {(InScript, COMCScript)}

δ(InScript,-->,COM) = {(InScript, e)}

f. Current state : InBody

δ(InBody,</body,CBody) = {(InBody, e)}

δ(InBody,>,BBody) = {(InHTML, e)}

δ(InBody,<link,CBody) = {(InBody, BLinkCBody)}

δ(InBody,rel,BLink) = {(InBody, Rel)}

δ(InBody,=,Rel) = {(InBody, =RelRel)}

δ(InBody,"str",=Rel) = {(InBody, e)}

δ(InBody,"nostr",=Rel) = {(InBody, e)}

δ(InBody,>,Rel) = {(InBody, e)}

δ(InBody,href,Rel) = {(InBody, HrefRel)}

δ(InBody,href,BLink) = {(InBody, HrefBLink)}

δ(InBody,=,Href) = {(InBody, =Href)}

δ(InBody,"str",=) = {(InBody, STR=)}

δ(InBody,"nostr",=) = {(InBody, NOSTR=)}

δ(InBody,e,Href) = {(InBody, e)}

δ(InBody,id,BLink) = {(InBody, idBLink)}

δ(InBody,class,BLink) = {(InBody, ClassBLink)}

δ(InBody,style,BLink) = {(InBody, StyleBLink)}

δ(InBody,id,Rel) = {(InBody, idRel)}

δ(InBody,class,Rel) = {(InBody, ClassRel)}

δ(InBody,style,Rel) = {(InBody, StyleRel)}

δ(InBody,<script,CBody) = {(InBody, BScriptCBody)}

δ(InBody,>,BScript) = {(InScript, CScriptBScript)}

δ(InBody,src,BScript) = {(InBody, SourceBScript)}

δ(InBody,=,Source) = {(InBody, =SourceSource)}

δ(InBody,"str",=Source) = {(InBody, STR=Source)}

δ(InBody,"nostr",=Source) = {(InBody, NOSTR=Source)}

δ(InBody,e,STR) = {(InBody, e)}

δ(InBody,e,=Source) = {(InBody, e)}

δ(InBody,e,Source) = {(InBody, e)}

δ(InBody,id,BScript) = {(InBody, idBScript)}

δ(InBody,class,BScript) = {(InBody, ClassBScript)}

δ(InBody,style,BScript) = {(InBody, StyleBScript)}

δ(InBody,<em,CBody) = {(InBody, BEmCBody)}

δ(InBody,>,BEm) = {(inEm, CEmBEm)}

δ(InBody,<b,CBody) = {(InBody, BBCBody)}

δ(InBody,>,BB) = {(inB, CBBB)}

δ(InBody,<abbr,CBody) = {(InBody, BAbbrCBody)}

δ(InBody,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InBody,<strong,CBody) = {(InBody, BStrongCBody)}

δ(InBody,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InBody,<small,CBody) = {(InBody, BSmallCBody)}

δ(InBody,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InBody,<h1,CBody) = {(InBody, BH1CBody)}

δ(InBody,id,BH1) = {(InBody, idBH1)}

δ(InBody,class,BH1) = {(InBody, ClassBH1)}

δ(InBody,style,BH1) = {(InBody, StyleBH1)}

δ(InBody,>,BH1) = {(InH1, CH1BH1)}

δ(InBody,<h2,CBody) = {(InBody, BH2CBody)}

δ(InBody,id,BH2) = {(InBody, idBH2)}

δ(InBody,class,BH2) = {(InBody, ClassBH2)}

δ(InBody,style,BH2) = {(InBody, StyleBH2)}

δ(InBody,>,BH2) = {(InH2, CH2BH2)}

δ(InBody,<h3,CBody) = {(InBody, BH3CBody)}

δ(InBody,id,BH3) = {(InBody, idBH3)}

δ(InBody,class,BH3) = {(InBody, ClassBH3)}

δ(InBody,style,BH3) = {(InBody, StyleBH3)}

δ(InBody,>,BH3) = {(InH3, CH3BH3)}

δ(InBody,<h4,CBody) = {(InBody, BH4CBody)}

δ(InBody,id,BH4) = {(InBody, idBH4)}

δ(InBody,class,BH4) = {(InBody, ClassBH4)}

δ(InBody,style,BH4) = {(InBody, StyleBH4)}

δ(InBody,>,BH4) = {(InH4, CH4BH4)}

δ(InBody,<h5,CBody) = {(InBody, BH5CBody)}

δ(InBody,id,BH5) = {(InBody, idBH5)}

δ(InBody,class,BH5) = {(InBody, ClassBH5)}

δ(InBody,style,BH5) = {(InBody, StyleBH5)}

δ(InBody,>,BH5) = {(InH5, CH5BH5)}

δ(InBody,<h6,CBody) = {(InBody, BH6CBody)}

δ(InBody,id,BH6) = {(InBody, idBH6)}

δ(InBody,class,BH6) = {(InBody, ClassBH6)}

δ(InBody,style,BH6) = {(InBody, StyleBH6)}

δ(InBody,>,BH6) = {(InH6, CH6BH6)}

δ(InBody,<table,CBody) = {(InBody, BTableCBody)}

δ(InBody,id,BTable) = {(InBody, idBTable)}

δ(InBody,class,BTable) = {(InBody, ClassBTable)}

δ(InBody,style,BTable) = {(InBody, styleBTable)}

δ(InBody,>,BTable) = {(InTable, CTableBTable)}

δ(InBody,=,id) = {(InBody, =id)}

δ(InBody,"str",=) = {(InBody, STR=)}

δ(InBody,"nostr",=) = {(InBody, NOSTR=)}

δ(InBody,e,STR) = {(InBody, e)}

δ(InBody,e,NOSTR) = {(InBody, e)}

δ(InBody,e,=) = {(, InBody)}

δ(InBody,e,id) = {(InBody, e)}

δ(InBody,=,Class) = {(InBody, =Class)}

δ(InBody,e,Class) = {(InBody, e)}

δ(InBody,=,Style) = {(InBody, =Style)}

δ(InBody,e,Style) = {(InBody, e)}

δ(InBody,<p,CBody) = {(InBody, BPCBody)}

δ(InBody,id,BP) = {(InBody, idBP)}

δ(InBody,class,BP) = {(InBody, ClassBP)}

δ(InBody,style,BP) = {(InBody, StyleBP)}

δ(InBody,>,BP) = {(InP, CPBP)}

δ(InBody,<br,CBody) = {(InBody, BRCBody)}

δ(InBody,id,BR) = {(InBody, idBR)}

δ(InBody,class,BR) = {(InBody, ClassBR)}

δ(InBody,style,BR) = {(InBody, StyleBR)}

δ(InBody,>,BR) = {(InBody, e)}

δ(InBody,<hr,CBody) = {(InBody, HRCBody)}

δ(InBody,id,HR) = {(InBody, idHR)}

δ(InBody,class,HR) = {(InBody, ClassHR)}

δ(InBody,style,HR) = {(InBody, StyleHR)}

δ(InBody,>,HR) = {(InBody, e)}

δ(InBody,<a,CBody) = {(InBody, BACBody)}

δ(InBody,id,BA) = {(InBody, idBA)}

δ(InBody,class,BA) = {(InBody, ClassBA)}

δ(InBody,style,BA) = {(InBody, StyleBA)}

δ(InBody,href,BA) = {(InBody, HrefBA)}

δ(InBody,=,Href) = {(InBody, =Href)}

δ(InBody,e,Href) = {(InBody, e)}

δ(InBody,>,BA) = {(InA, CABA)}

δ(InBody,<img,CBody) = {(InBody, ImgCBody)}

δ(InBody,src,Img) = {(InBody, Src)}

δ(InBody,=,Src) = {(InBody, =SrcSrc)}

δ(InBody,"str",=Src) = {(InBody, e)}

δ(InBody,"nostr",=Src) = {(InBody, e)}

δ(InBody,>,Src) = {(InBody, e)}

δ(InBody,alt,Src) = {(InBody, AltSrc)}

δ(InBody,alt,Img) = {(InBody, AltImg)}

δ(InBody,=,Alt) = {(InBody, =Alt)}

δ(InBody,e,Alt) = {(InBody, e)}

δ(InBody,<input,CBody) = {(InBody, BInputCBody)}

δ(InBody,id,BInput) = {(InBody, idBInput)}

δ(InBody,class,BInput) = {(InBody, ClassBInput)}

δ(InBody,style,BInput) = {(InBody, StyleBInput)}

δ(InBody,type,BInput) = {(InBody, TypeBInput)}

δ(InBody,=,type) = {(InBody, =TypeType)}

δ(InBody,"text",=Type) = {(InBody, STR=Type)}

δ(InBody,"password",=Type) = {(InBody, STR=Type)}

δ(InBody,"email",=Type) = {(InBody, STR=Type)}

δ(InBody,"number",=Type) = {(InBody, STR=Type)}

δ(InBody,"checkbox",=Type) = {(InBody, STR=Type)}

δ(InBody,e,=Type) = {(InBody, e)}

δ(InBody,e,Type) = {(InBody, e)}

δ(InBody,>,BInput) = {(InBody, e)}

δ(InBody,<button,CBody) = {(InBody, BButtonCBody)}

δ(InBody,type,BButton) = {(InBody, TypeBButton)}

δ(InBody,=,Type) = {(InBody, =TypeType)}

δ(InBody,"submit",=Type) = {(InBody, STR=Type)}

δ(InBody,"reset",=Type) = {(InBody, STR=Type)}

δ(InBody,"button",=Type) = {(InBody, STR=Type)}

δ(InBody,id,BButton) = {(InBody, idBButton)}

δ(InBody,class,BButton) = {(InBody, ClassBUBButton)}

δ(InBody,style,BButton) = {(InBody, StyleBButton)}

δ(InBody,>,BButton) = {(InButton, CButtonBButton)}

δ(InBody,<!--,CBody) = {(InBody, COMCBody)}

δ(InBody,-->,COM) = {(InBody, e)}

δ(InBody,<form,CBody) = {(InBody, BFormCBody)}

δ(InBody,id,BForm) = {(InBody, idBForm)}

δ(InBody,class,BForm) = {(InBody, ClassBForm)}

δ(InBody,style,BForm) = {(InBody, StyleBForm)}

δ(InBody,action,BForm) = {(InBody, ACTBForm)}

δ(InBody,=,ACT) = {(InBody, =ACT)}

δ(InBody,"str",=) = {(InBody, STR=)}

δ(InBody,"nostr",=) = {(InBody, NOSTR=)}

δ(InBody,e,ACT) = {(InBody, e)}

δ(InBody,method,BForm) = {(InBody, METHODBForm)}

δ(InBody,=,METHOD) = {(InBody, =)}

δ(InBody,"GET",=) = {(InBody, e)}

δ(InBody,"POST",=) = {(InBody, e)}

δ(InBody,>,BForm) = {(InDiv, Form)}

δ(InBody,<div,CBody) = {(InBody, BDivCBody)}

δ(InBody,id,BDiv) = {(InBody, idBDiv)}

δ(InBody,class,BDiv) = {(InBody, ClassBDiv)}

δ(InBody,style,BDiv) = {(InBody, StyleBDiv)}

δ(InBody,>,BDiv) = {(InDiv, Div)}

g. Current state : InH1

δ(InH1,str,CH1) = {(InH1, CH1)}

δ(InH1,</h1,CH1) = {(InH1, e)}

δ(InH1,>,BH1) = {(InH1, e)}

δ(InH1,e,CBody) = {(InBody, CBody)}

δ(InH1,e,Div) = {(InDiv, Div)}

δ(InH1,e,Form) = {(InDiv, Form)}

δ(InH1,<em,CH1) = {(InH1, BEmCH1)}

δ(InH1,>,BEm) = {(inEm, CEmBEm)}

δ(InH1,<b,CH1) = {(InH1, BBCH1)}

δ(InH1,>,BB) = {(inB, CBBB)}

δ(InH1,<abbr,CH1) = {(InH1, BAbbrCH1)}

δ(InH1,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InH1,<strong,CH1) = {(InH1, BStrongCH1)}

δ(InH1,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InH1,<small,CH1) = {(InH1, BSmallCH1)}

δ(InH1,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InH1,<!--,CH1) = {(InH1, COMCH1)}

δ(InH1,-->,COM) = {(InH1, e)}

h. Current state : InH2

δ(InH2,str,CH2) = {(InH2, CH2)}

δ(InH2,</h2,CH2) = {(InH2, e)}

δ(InH2,>,BH2) = {(InH2, e)}

δ(InH2,e,CBody) = {(InBody, CBody)}

δ(InH2,e,Div) = {(InDiv, Div)}

δ(InH2,e,Form) = {(InDiv, Form)}

δ(InH2,<em,CH2) = {(InH2, BEmCH2)}

δ(InH2,>,BEm) = {(inEm, CEmBEm)}

δ(InH2,<b,CH2) = {(InH2, BBCH2)}

δ(InH2,>,BB) = {(inB, CBBB)}

δ(InH2,<abbr,CH2) = {(InH2, BAbbrCH2)}

δ(InH2,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InH2,<strong,CH2) = {(InH2, BStrongCH2)}

δ(InH2,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InH2,<small,CH2) = {(InH2, BSmallCH2)}

δ(InH2,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InH2,<!--,CH2) = {(InH2, COMForm)}

δ(InH2,-->,COM) = {(InH2, e)}

i. Current state : InH3

δ(InH3,str,CH3) = {(InH3, CH3)}

δ(InH3,</h3,CH3) = {(InH3, e)}

δ(InH3,>,BH3) = {(InH3, e)}

δ(InH3,e,CBody) = {(InBody, CBody)}

δ(InH3,e,Div) = {(InDiv, Div)}

δ(InH3,e,Form) = {(InDiv, Form)}

δ(InH3,<em,CH3) = {(InH3, BEmCH3)}

δ(InH3,>,BEm) = {(inEm, CEmBEm)}

δ(InH3,<b,CH3) = {(InH3, BBCH3)}

δ(InH3,>,BB) = {(inB, CBBB)}

δ(InH3,<abbr,CH3) = {(InH3, BAbbrCH3)}

δ(InH3,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InH3,<strong,CH3) = {(InH3, BStrongCH3)}

δ(InH3,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InH3,<small,CH3) = {(InH3, BSmallCH3)}

δ(InH3,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InH3,<!--,CH3) = {(InH3, COMCH3)}

δ(InH3,-->,COM) = {(InH3, e)}

j. Current state : InH4

δ(InH4,str,CH4) = {(InH4, CH4)}

δ(InH4,</h4,CH4) = {(InH4, e)}

δ(InH4,>,BH4) = {(InH4, e)}

δ(InH4,e,CBody) = {(InBody, CBody)}

δ(InH4,e,Div) = {(InDiv, Div)}

δ(InH4,e,Form) = {(InDiv, Form)}

δ(InH4,<em,CH4) = {(InH4, BEmCH4)}

δ(InH4,>,BEm) = {(inEm, CEmBEm)}

δ(InH4,<b,CH4) = {(InH4, BBCH4)}

δ(InH4,>,BB) = {(inB, CBBB)}

δ(InH4,<abbr,CH4) = {(InH4, BAbbrCH4)}

δ(InH4,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InH4,<strong,CH4) = {(InH4, BStrongCH4)}

δ(InH4,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InH4,<small,CH4) = {(InH4, BSmallCH4)}

δ(InH4,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InH4,<!--,CH4) = {(InH4, COMCH4)}

δ(InH4,-->,COM) = {(InH4, e)}

k. Current state : InH5

δ(InH5,<em,CH5) = {(InH5, BEmCH5)}

δ(InH5,>,BEm) = {(inEm, CEmBEm)}

δ(InH5,<b,CH5) = {(InH5, BBCH5)}

δ(InH5,>,BB) = {(inB, CBBB)}

δ(InH5,<abbr,CH5) = {(InH5, BAbbrCH5)}

δ(InH5,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InH5,<strong,CH5) = {(InH5, BStrongCH5)}

δ(InH5,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InH5,<small,CH5) = {(InH5, BSmallCH5)}

δ(InH5,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InH5,str,CH5) = {(InH5, CH5)}

δ(InH5,</h5,CH5) = {(InH5, e)}

δ(InH5,>,BH5) = {(InH5, e)}

δ(InH5,e,CBody) = {(InBody, CBody)}

δ(InH5,e,Div) = {(InDiv, Div)}

δ(InH5,e,Form) = {(InDiv, Form)}

δ(InH5,<!--,CH5) = {(InH5, COMCH5)}

δ(InH5,-->,COM) = {(InH5, e)}

l. Current state : InH6

δ(InH6,str,CH6) = {(InH6, CH6)}

δ(InH6,</h6,CH6) = {(InH6, e)}

δ(InH6,>,BH6) = {(InH6, e)}

δ(InH6,e,CBody) = {(InBody, CBody)}

δ(InH6,e,Div) = {(InDiv, Div)}

δ(InH6,e,Form) = {(InDiv, Form)}

δ(InH6,<em,CH6) = {(InH6, BEmCH6)}

δ(InH6,>,BEm) = {(inEm, CEmBEm)}

δ(InH6,<b,CH6) = {(InH6, BBCH6)}

δ(InH6,>,BB) = {(inB, CBBB)}

δ(InH6,<abbr,CH6) = {(InH6, BAbbrCH6)}

δ(InH6,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InH6,<strong,CH6) = {(InH6, BStrongCH6)}

δ(InH6,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InH6,<small,CH6) = {(InH6, BSmallCH6)}

δ(InH6,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InH6,<!--,CH6) = {(InH6, COMCH6)}

δ(InH6,-->,COM) = {(InH6, e)}

m. Current state : InA

δ(InA,str,CA) = {(InA, CA)}

δ(InA,</a,CA) = {(InA, e)}

δ(InA,>,BA) = {(InA, e)}

δ(InA,e,CBody) = {(InBody, CBody)}

δ(InA,e,Div) = {(InDiv, Div)}

δ(InA,e,Form) = {(InDiv, Form)}

δ(InA,<em,CA) = {(InA, BEmCA)}

δ(InA,>,BEm) = {(inEm, CEmBEm)}

δ(InA,<b,CA) = {(InA, BBCA)}

δ(InA,>,BB) = {(inB, CBBB)}

δ(InA,<abbr,CA) = {(InA, BAbbrCA)}

δ(InA,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InA,<strong,CA) = {(InA, BStrongCA)}

δ(InA,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InA,<small,CA) = {(InA, BSmallCA)}

δ(InA,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InA,<!--,CA) = {(InA, COMCA)}

δ(InA,-->,COM) = {(InA, e)}

n. Current state : InP

δ(InP,str,CP) = {(InP, CP)}

δ(InP,</p,CP) = {(InP, e)}

δ(InP,>,BP) = {(InP, e)}

δ(InP,e,CBody) = {(InBody, CBody)}

δ(InP,e,Div) = {(InDiv, Div)}

δ(InP,e,Form) = {(InDiv, Form)}

δ(InP,<em,CP) = {(InP, BEmCP)}

δ(InP,>,BEm) = {(inEm, CEmBEm)}

δ(InP,<b,CP) = {(InP, BBCP)}

δ(InP,>,BB) = {(inB, CBBB)}

δ(InP,<abbr,CP) = {(InP, BAbbrCP)}

δ(InP,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InP,<strong,CP) = {(InP, BStrongCP)}

δ(InP,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InP,<small,CP) = {(InP, BSmallCP)}

δ(InP,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InP,<!--,CP) = {(InP, COMCP)}

δ(InP,-->,COM) = {(InP, e)}

o. Current state : inEm

δ(inEm,str,CEm) = {(inEm, CEm)}

δ(inEm,</em,CEm) = {(inEm, e)}

δ(inEm,>,BEm) = {(inEm, e)}

δ(inEm,e,CTitle) = {(InTitle, CTitle)}

δ(inEm,e,CScript) = {(InScript, CScript)}

δ(inEm,e,CH1) = {(InH1, CH1)}

δ(inEm,e,CH2) = {(InH2, CH2)}

δ(inEm,e,CH3) = {(InH3, CH3)}

δ(inEm,e,CH4) = {(InH4, CH4)}

δ(inEm,e,CH5) = {(InH5, CH5)}

δ(inEm,e,CH6) = {(InH6, CH6)}

δ(inEm,e,CP) = {(InP, CP)}

δ(inEm,e,Div) = {(InDiv, Div)}

δ(inEm,e,Form) = {(InDiv, Form)}

δ(inEm,e,CA) = {(InA, CA)}

δ(inEm,e,CButton) = {(InButton, CButton)}

δ(inEm,e,CTD) = {(InTD, CTD)}

δ(inEm,e,CTH) = {(InTH, CTH)}

δ(inEm,e,CBody) = {(InBody, CBody)}

δ(inEm,<!--,CEm) = {(inEm, COMCEm)}

δ(inEm,-->,COM) = {(inEm, e)}

p. Current state : inB

δ(inB,str,CB) = {(inB, CB)}

δ(inB,</b,CB) = {(inB, e)}

δ(inB,>,BB) = {(inB, e)}

δ(inB,e,CTitle) = {(InTitle, CTitle)}

δ(inB,e,CScript) = {(InScript, CScript)}

δ(inB,e,CH1) = {(InH1, CH1)}

δ(inB,e,CH2) = {(InH2, CH2)}

δ(inB,e,CH3) = {(InH3, CH3)}

δ(inB,e,CH4) = {(InH4, CH4)}

δ(inB,e,CH5) = {(InH5, CH5)}

δ(inB,e,CH6) = {(InH6, CH6)}

δ(inB,e,CP) = {(InP, CP)}

δ(inB,e,Div) = {(InDiv, Div)}

δ(inB,e,Form) = {(InDiv, Form)}

δ(inB,e,CA) = {(InA, CA)}

δ(inB,e,CButton) = {(InButton, CButton)}

δ(inB,e,CTD) = {(InTD, CTD)}

δ(inB,e,CTH) = {(InTH, CTH)}

δ(inB,e,CBody) = {(InBody, CBody)}

δ(inB,<!--,CB) = {(inB, COMCB)}

δ(inB,-->,COM) = {(inB, e)}

q. Current state : inAbbr

δ(inAbbr,str,CAbbr) = {(inAbbr, CAbbr)}

δ(inAbbr,</abbr,CAbbr) = {(inAbbr, e)}

δ(inAbbr,>,BAbbr) = {(inAbbr, e)}

δ(inAbbr,e,CTitle) = {(InTitle, CTitle)}

δ(inAbbr,e,CScript) = {(InScript, CScript)}

δ(inAbbr,e,CH1) = {(InH1, CH1)}

δ(inAbbr,e,CH2) = {(InH2, CH2)}

δ(inAbbr,e,CH3) = {(InH3, CH3)}

δ(inAbbr,e,CH4) = {(InH4, CH4)}

δ(inAbbr,e,CH5) = {(InH5, CH5)}

δ(inAbbr,e,CH6) = {(InH6, CH6)}

δ(inAbbr,e,CP) = {(InP, CP)}

δ(inAbbr,e,Div) = {(InDiv, Div)}

δ(inAbbr,e,Form) = {(InDiv, Form)}

δ(inAbbr,e,CA) = {(InA, CA)}

δ(inAbbr,e,CButton) = {(InButton, CButton)}

δ(inAbbr,e,CTD) = {(InTD, CTD)}

δ(inAbbr,e,CTH) = {(InTH, CTH)}

δ(inAbbr,e,CBody) = {(InBody, CBody)}

δ(inAbbr,<!--,CAbbr) = {(inAbbr, COMCAbbr)}

δ(inAbbr,-->,COM) = {(inAbbr, e)}

r. Current state : inSmall

δ(inSmall,str,CSmall) = {(inSmall, CSmall)}

δ(inSmall,</small,CSmall) = {(inSmall, e)}

δ(inSmall,>,BSmall) = {(inSmall, e)}

δ(inSmall,e,CTitle) = {(InTitle, CTitle)}

δ(inSmall,e,CScript) = {(InScript, CScript)}

δ(inSmall,e,CH1) = {(InH1, CH1)}

δ(inSmall,e,CH2) = {(InH2, CH2)}

δ(inSmall,e,CH3) = {(InH3, CH3)}

δ(inSmall,e,CH4) = {(InH4, CH4)}

δ(inSmall,e,CH5) = {(InH5, CH5)}

δ(inSmall,e,CH6) = {(InH6, CH6)}

δ(inSmall,e,CP) = {(InP, CP)}

δ(inSmall,e,Div) = {(InDiv, Div)}

δ(inSmall,e,Form) = {(InDiv, Form)}

δ(inSmall,e,CA) = {(InA, CA)}

δ(inSmall,e,CButton) = {(InButton, CButton)}

δ(inSmall,e,CTD) = {(InTD, CTD)}

δ(inSmall,e,CTH) = {(InTH, CTH)}

δ(inSmall,e,CBody) = {(InBody, CBody)}

δ(inSmall,<!--,CSmall) = {(inSmall, COMCSmall)}

δ(inSmall,-->,COM) = {(inSmall, e)}

s. Current state : inStrong

δ(inStrong,str,CStrong) = {(inStrong, CStrong)}

δ(inStrong,</strong,CStrong) = {(inStrong, e)}

δ(inStrong,>,BStrong) = {(inStrong, e)}

δ(inStrong,e,CTitle) = {(InTitle, CTitle)}

δ(inStrong,e,CScript) = {(InScript, CScript)}

δ(inStrong,e,CH1) = {(InH1, CH1)}

δ(inStrong,e,CH2) = {(InH2, CH2)}

δ(inStrong,e,CH3) = {(InH3, CH3)}

δ(inStrong,e,CH4) = {(InH4, CH4)}

δ(inStrong,e,CH5) = {(InH5, CH5)}

δ(inStrong,e,CH6) = {(InH6, CH6)}

δ(inStrong,e,CP) = {(InP, CP)}

δ(inStrong,e,Div) = {(InDiv, Div)}

δ(inStrong,e,Form) = {(InDiv, Form)}

δ(inStrong,e,CA) = {(InA, CA)}

δ(inStrong,e,CButton) = {(InButton, CButton)}

δ(inStrong,e,CTD) = {(InTD, CTD)}

δ(inStrong,e,CTH) = {(InTH, CTH)}

δ(inStrong,e,CBody) = {(InBody, CBody)}

δ(inStrong,<!--,CStrong) = {(inStrong, COMCStrong)}

δ(inStrong,-->,COM) = {(inStrong, e)}

t. Current state : InTable

δ(InTable,<tr,CTable) = {(InTable, BTRCTable)}

δ(InTable,id,BTR) = {(InTable, idBTR)}

δ(InTable,class,BTR) = {(InTable, ClassBTR)}

δ(InTable,style,BTR) = {(InTable, StyleBTR)}

δ(InTable,>,BTR) = {(InTR, CTRBTR)}

δ(InTable,=,id) = {(InTable, =id)}

δ(InTable,"str",=) = {(InTable, STR=)}

δ(InTable,"nostr",=) = {(InTable, NOSTR=)}

δ(InTable,e,STR) = {(InTable, e)}

δ(InTable,e,NOSTR) = {(InTable, e)}

δ(InTable,e,=) = {(, InTable)}

δ(InTable,e,id) = {(InTable, e)}

δ(InTable,=,Class) = {(InTable, =Class)}

δ(InTable,e,Class) = {(InTable, e)}

δ(InTable,=,Style) = {(InTable, =Style)}

δ(InTable,e,Style) = {(InTable, e)}

δ(InTable,</table,CTable) = {(InTable, e)}

δ(InTable,>,BTable) = {(InTable, e)}

δ(InTable,e,CBody) = {(InBody, CBody)}

δ(InTable,e,Div) = {(InDiv, Div)}

δ(InTable,e,Form) = {(InDiv, Form)}

δ(InTable,<!--,CTable) = {(InTable, COMCTable)}

δ(InTable,-->,COM) = {(InTable, e)}

u. Current state : InTR

δ(InTR,</tr,CTR) = {(InTR, e)}

δ(InTR,>,BTR) = {(InTable, e)}

δ(InTR,<td,CTR) = {(InTR, BTDCTR)}

δ(InTR,id,BTD) = {(InTR, idBTD)}

δ(InTR,style,BTD) = {(InTR, StyleBTD)}

δ(InTR,class,BTD) = {(InTR, ClassBTD)}

δ(InTR,>,BTD) = {(InTD, CTDBTD)}

δ(InTR,<th,CTR) = {(InTR, BTHCTR)}

δ(InTR,id,BTH) = {(InTR, idBTH)}

δ(InTR,style,BTH) = {(InTR, StyleBTH)}

δ(InTR,class,BTH) = {(InTR, ClassBTH)}

δ(InTR,>,BTH) = {(InTH, CTHBTH)}

δ(InTR,=,id) = {(InTR, =id)}

δ(InTR,"str",=) = {(InTR, STR=)}

δ(InTR,"nostr",=) = {(InTR, NOSTR=)}

δ(InTR,e,STR) = {(InTR, e)}

δ(InTR,e,NOSTR) = {(InTR, e)}

δ(InTR,e,=) = {(, InTR)}

δ(InTR,e,id) = {(InTR, e)}

δ(InTR,=,Class) = {(InTR, =Class)}

δ(InTR,e,Class) = {(InTR, e)}

δ(InTR,=,Style) = {(InTR, =Style)}

δ(InTR,e,Style) = {(InTR, e)}

δ(InTR,<!--,CTR) = {(InTR, COMCTR)}

δ(InTR,-->,COM) = {(InTR, e)}

v. Current state : InTD

δ(InTD,str,CTD) = {(InTD, CTD)}

δ(InTD,</td,CTD) = {(InTD, e)}

δ(InTD,>,BTD) = {(InTD, e)}

δ(InTD,e,CTable) = {(InTable, CTable)}

δ(InTD,e,CTR) = {(InTR, CTR)}

δ(InTD,<em,CTD) = {(InTD, BEmCTD)}

δ(InTD,>,BEm) = {(inEm, CEmBEm)}

δ(InTD,<b,CTD) = {(InTD, BBCTD)}

δ(InTD,>,BB) = {(inB, CBBB)}

δ(InTD,<abbr,CTD) = {(InTD, BAbbrCTD)}

δ(InTD,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InTD,<strong,CTD) = {(InTD, BStrongCTD)}

δ(InTD,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InTD,<small,CTD) = {(InTD, BSmallCTD)}

δ(InTD,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InTD,<!--,CTD) = {(InTD, COMCTD)}

δ(InTD,-->,COM) = {(InTD, e)}

w. **Current state : InTH**

δ(InTH,str,CTH) = {(InTH, CTH)}

δ(InTH,</th,CTH) = {(InTH, e)}

δ(InTH,>,BTH) = {(InTH, e)}

δ(InTH,e,CTR) = {(InTR, CTR)}

δ(InTH,e,CTable) = {(InTable, CTable)}

δ(InTH,<em,CTH) = {(InTH, BEmCTH)}

δ(InTH,>,BEm) = {(inEm, CEmBEm)}

δ(InTH,<b,CTH) = {(InTH, BBCTH)}

δ(InTH,>,BB) = {(inB, CBBB)}

δ(InTH,<abbr,CTH) = {(InTH, BAbbrCTH)}

δ(InTH,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InTH,<strong,CTH) = {(InTH, BStrongCTH)}

δ(InTH,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InTH,<small,CTH) = {(InTH, BSmallCTH)}

δ(InTH,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InTH,<!--,CTH) = {(InTH, COMCTH)}

δ(InTH,-->,COM) = {(InTH, e)}

x. **Current state : InButton**

δ(InButton,str,CButton) = {(InButton, CButton)}

δ(InButton,</button,CButton) = {(InButton, e)}

δ(InButton,>,BButton) = {(InButton, e)}

δ(InButton,e,CBody) = {(InBody, CBody)}

δ(InButton,e,Div) = {(InDiv, Div)}

δ(InButton,e,Form) = {(InDiv, Form)}

δ(InButton,<em,CButton) = {(InButton, BEmCButton)}

δ(InButton,>,BEm) = {(inEm, CEmBEm)}

δ(InButton,<b,CButton) = {(InButton, BBCButton)}

δ(InButton,>,BB) = {(inB, CBBB)}

δ(InButton,<abbr,CButton) = {(InButton, BAbbrCButton)}

δ(InButton,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InButton,<strong,CButton) = {(InButton, BStrongCButton)}

δ(InButton,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InButton,<small,CButton) = {(InButton, BSmallCButton)}

δ(InButton,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InButton,<!--,CButton) = {(InButton, COMCButton)}

δ(InButton,-->,COM) = {(InButton, e)}

y. Current state : InDiv

δ(InDiv,str,Div) = {(InDiv, Div)}

δ(InDiv,str,Form) = {(InDiv, Form)}

δ(InDiv,</div,Div) = {(InDiv, CDiv)}

δ(InDiv,>,CDiv) = {(InDiv, e)}

δ(InDiv,</form,Form) = {(InDiv, CForm)}

δ(InDiv,>,CForm) = {(InDiv, e)}

δ(InDiv,e,CBody) = {(InBody, CBody)}

δ(InDiv,<em,Div) = {(InDiv, BEmDiv)}

δ(InDiv,<em,Form) = {(InDiv, BEmForm)}

δ(InDiv,>,BEm) = {(inEm, CEmBEm)}

δ(InDiv,<b,Div) = {(InDiv, BBDiv)}

δ(InDiv,<b,Form) = {(InDiv, BBForm)}

δ(InDiv,>,BB) = {(inB, CBBB)}

δ(InDiv,<abbr,Div) = {(InDiv, BAbbrDiv)}

δ(InDiv,<abbr,Form) = {(InDiv, BAbbrForm)}

δ(InDiv,>,BAbbr) = {(inAbbr, CAbbrBAbbr)}

δ(InDiv,<strong,Div) = {(InDiv, BStrongDiv)}

δ(InDiv,<strong,Form) = {(InDiv, BStrongForm)}

δ(InDiv,>,BStrong) = {(inStrong, CStrongBStrong)}

δ(InDiv,<small,Div) = {(InDiv, BSmallDiv)}

δ(InDiv,<small,Form) = {(InDiv, BSmallForm)}

δ(InDiv,>,BSmall) = {(inSmall, CSmallBSmall)}

δ(InDiv,<h1,Div) = {(InDiv, BH1Div)}

δ(InDiv,<h1,Form) = {(InDiv, BH1Form)}

δ(InDiv,id,BH1) = {(InDiv, idBH1)}

δ(InDiv,class,BH1) = {(InDiv, ClassBH1)}

δ(InDiv,style,BH1) = {(InDiv, StyleBH1)}

δ(InDiv,>,BH1) = {(InH1, CH1BH1)}

δ(InDiv,<h2,Div) = {(InDiv, BH2Div)}

δ(InDiv,<h2,Form) = {(InDiv, BH2Form)}

δ(InDiv,id,BH2) = {(InDiv, idBH2)}

δ(InDiv,class,BH2) = {(InDiv, ClassBH2)}

δ(InDiv,style,BH2) = {(InDiv, StyleBH2)}

δ(InDiv,>,BH2) = {(InH2, CH2BH2)}

δ(InDiv,<h3,Div) = {(InDiv, BH3Div)}

δ(InDiv,<h3,Form) = {(InDiv, BH3Form)}

δ(InDiv,id,BH3) = {(InDiv, idBH3)}

δ(InDiv,class,BH3) = {(InDiv, ClassBH3)}

δ(InDiv,style,BH3) = {(InDiv, StyleBH3)}

δ(InDiv,>,BH3) = {(InH3, CH3BH3)}

δ(InDiv,<h4,Div) = {(InDiv, BH4Div)}

δ(InDiv,<h4,Form) = {(InDiv, BH4Form)}

δ(InDiv,id,BH4) = {(InDiv, idBH4)}

δ(InDiv,class,BH4) = {(InDiv, ClassBH4)}

δ(InDiv,style,BH4) = {(InDiv, StyleBH4)}

δ(InDiv,>,BH4) = {(InH4, CH4BH4)}

δ(InDiv,<h5,Div) = {(InDiv, BH5Div)}

δ(InDiv,<h5,Form) = {(InDiv, BH5Form)}

δ(InDiv,id,BH5) = {(InDiv, idBH5)}

δ(InDiv,class,BH5) = {(InDiv, ClassBH5)}

δ(InDiv,style,BH5) = {(InDiv, StyleBH5)}

δ(InDiv,>,BH5) = {(InH5, CH5BH5)}

δ(InDiv,<h6,Div) = {(InDiv, BH6Div)}

δ(InDiv,<h6,Form) = {(InDiv, BH6Form)}

δ(InDiv,id,BH6) = {(InDiv, idBH6)}

δ(InDiv,class,BH6) = {(InDiv, ClassBH6)}

δ(InDiv,style,BH6) = {(InDiv, StyleBH6)}

δ(InDiv,>,BH6) = {(InH6, CH6BH6)}

δ(InDiv,<table,Div) = {(InDiv, BTableDiv)}

δ(InDiv,<table,Form) = {(InDiv, BTableForm)}

δ(InDiv,id,BTable) = {(InDiv, idBTable)}

δ(InDiv,class,BTable) = {(InDiv, ClassBTable)}

δ(InDiv,style,BTable) = {(InDiv, styleBTable)}

δ(InDiv,>,BTable) = {(InTable, CTableBTable)}

δ(InDiv,=,id) = {(InDiv, =id)}

δ(InDiv,"str",=) = {(InDiv, STR=)}

δ(InDiv,"nostr",=) = {(InDiv, NOSTR=)}

δ(InDiv,e,STR) = {(InDiv, e)}

δ(InDiv,e,NOSTR) = {(InDiv, e)}

δ(InDiv,e,=) = {(, InDiv)}

δ(InDiv,e,id) = {(InDiv, e)}

δ(InDiv,=,Class) = {(InDiv, =Class)}

δ(InDiv,e,Class) = {(InDiv, e)}

δ(InDiv,=,Style) = {(InDiv, =Style)}

δ(InDiv,e,Style) = {(InDiv, e)}

δ(InDiv,<p,Div) = {(InDiv, BPDiv)}

δ(InDiv,<p,Form) = {(InDiv, BPForm)}

δ(InDiv,id,BP) = {(InDiv, idBP)}

δ(InDiv,class,BP) = {(InDiv, ClassBP)}

δ(InDiv,style,BP) = {(InDiv, StyleBP)}

δ(InDiv,>,BP) = {(InP, CPBP)}

δ(InDiv,<br,Div) = {(InDiv, BRDiv)}

δ(InDiv,<br,Form) = {(InDiv, BRForm)}

δ(InDiv,id,BR) = {(InDiv, idBR)}

δ(InDiv,class,BR) = {(InDiv, ClassBR)}

δ(InDiv,style,BR) = {(InDiv, StyleBR)}

δ(InDiv,>,BR) = {(InDiv, e)}

δ(InDiv,<hr,Div) = {(InDiv, HRDiv)}

δ(InDiv,<hr,Form) = {(InDiv, HRForm)}

δ(InDiv,id,HR) = {(InDiv, idHR)}

δ(InDiv,class,HR) = {(InDiv, ClassHR)}

δ(InDiv,style,HR) = {(InDiv, StyleHR)}

δ(InDiv,>,HR) = {(InDiv, e)}

δ(InDiv,<a,Div) = {(InDiv, BADiv)}

δ(InDiv,<a,Form) = {(InDiv, BAForm)}

δ(InDiv,id,BA) = {(InDiv, idBA)}

δ(InDiv,class,BA) = {(InDiv, ClassBA)}

δ(InDiv,style,BA) = {(InDiv, StyleBA)}

δ(InDiv,href,BA) = {(InDiv, HrefBA)}

δ(InDiv,=,Href) = {(InDiv, =Href)}

δ(InDiv,e,Href) = {(InDiv, e)}

δ(InDiv,>,BA) = {(InA, CABA)}

δ(InDiv,<img,Div) = {(InDiv, ImgDiv)}

δ(InDiv,<img,Form) = {(InDiv, ImgForm)}

δ(InDiv,src,Img) = {(InDiv, Src)}

δ(InDiv,=,Src) = {(InDiv, =SrcSrc)}

δ(InDiv,"str",=Src) = {(InDiv, e)}

δ(InDiv,"nostr",=Src) = {(InDiv, e)}

δ(InDiv,>,Src) = {(InDiv, e)}

δ(InDiv,alt,Src) = {(InDiv, AltSrc)}

δ(InDiv,alt,Img) = {(InDiv, AltImg)}

δ(InDiv,=,Alt) = {(InDiv, =Alt)}

δ(InDiv,e,Alt) = {(InDiv, e)}

δ(InDiv,<input,Div) = {(InDiv, BInputDiv)}

δ(InDiv,<input,Form) = {(InDiv, BInputForm)}

δ(InDiv,id,BInput) = {(InDiv, idBInput)}

δ(InDiv,class,BInput) = {(InDiv, ClassBInput)}

δ(InDiv,style,BInput) = {(InDiv, StyleBInput)}

δ(InDiv,type,BInput) = {(InDiv, TypeBInput)}

δ(InDiv,=,type) = {(InDiv, =TypeType)}

δ(InDiv,"text",=Type) = {(InDiv, STR=Type)}

δ(InDiv,"password",=Type) = {(InDiv, STR=Type)}

δ(InDiv,"email",=Type) = {(InDiv, STR=Type)}

δ(InDiv,"number",=Type) = {(InDiv, STR=Type)}

δ(InDiv,"checkbox",=Type) = {(InDiv, STR=Type)}

δ(InDiv,e,=Type) = {(InDiv, e)}

δ(InDiv,e,Type) = {(InDiv, e)}

δ(InDiv,>,BInput) = {(InDiv, e)}

δ(InDiv,<button,Div) = {(InDiv, BButtonDiv)}

δ(InDiv,<button,Form) = {(InDiv, BButtonForm)}

δ(InDiv,type,BButton) = {(InDiv, TypeBButton)}

δ(InDiv,=,Type) = {(InDiv, =TypeType)}

δ(InDiv,"submit",=Type) = {(InDiv, STR=Type)}

δ(InDiv,"reset",=Type) = {(InDiv, STR=Type)}

δ(InDiv,"button",=Type) = {(InDiv, STR=Type)}

δ(InDiv,id,BButton) = {(InDiv, idBButton)}

δ(InDiv,class,BButton) = {(InDiv, ClassBUBButton)}

δ(InDiv,style,BButton) = {(InDiv, StyleBButton)}

δ(InDiv,>,BButton) = {(InButton, CButtonBButton)}

δ(InDiv,<!--,Div) = {(InDiv, COMDiv)}

δ(InDiv,<!--,Form) = {(InDiv, COMForm)}

δ(InDiv,-->,COM) = {(InDiv, e)}

δ(InDiv,<form,Div) = {(InDiv, BFormDiv)}

δ(InDiv,id,BForm) = {(InDiv, idBForm)}

δ(InDiv,class,BForm) = {(InDiv, ClassBForm)}

δ(InDiv,style,BForm) = {(InDiv, StyleBForm)}

δ(InDiv,action,BForm) = {(InDiv, ACTBForm)}

δ(InDiv,=,ACT) = {(InDiv, =ACT)}

δ(InDiv,"str",=) = {(InDiv, STR=)}

δ(InDiv,"nostr",=) = {(InDiv, NOSTR=)}

δ(InDiv,e,ACT) = {(InDiv, e)}

δ(InDiv,method,BForm) = {(InDiv, METHODBForm)}

δ(InDiv,=,METHOD) = {(InDiv, =)}

δ(InDiv,"GET",=) = {(InDiv, e)}

δ(InDiv,"POST",=) = {(InDiv, e)}

δ(InDiv,>,BForm) = {(InDiv, Form)}

δ(InDiv,<div,Div) = {(InDiv, BDivDiv)}

δ(InDiv,<div,Form) = {(InDiv, BDivForm)}

δ(InDiv,id,BDiv) = {(InDiv, idBDiv)}

δ(InDiv,class,BDiv) = {(InDiv, ClassBDiv)}

δ(InDiv,style,BDiv) = {(InDiv, StyleBDiv)}

δ(InDiv,>,BDiv) = {(InDiv, Div)}

δ(InDiv,<link,CScript) = {(InDiv, BLinkCScript)}

δ(InDiv,<link,Div) = {(InDiv, BLinkDiv)}

δ(InDiv,rel,BLink) = {(InDiv, Rel)}

δ(InDiv,=,Rel) = {(InDiv, =RelRel)}

δ(InDiv,"str",=Rel) = {(InDiv, e)}

δ(InDiv,"nostr",=Rel) = {(InDiv, e)}

δ(InDiv,>,Rel) = {(InDiv, e)}

δ(InDiv,href,Rel) = {(InDiv, HrefRel)}

δ(InDiv,href,BLink) = {(InDiv, HrefBLink)}

δ(InDiv,=,Href) = {(InDiv, =Href)}

δ(InDiv,"str",=) = {(InDiv, STR=)}

δ(InDiv,"nostr",=) = {(InDiv, NOSTR=)}

δ(InDiv,e,Href) = {(InDiv, e)}

δ(InDiv,id,BLink) = {(InDiv, idBLink)}

δ(InDiv,class,BLink) = {(InDiv, ClassBLink)}

δ(InDiv,style,BLink) = {(InDiv, StyleBLink)}

δ(InDiv,id,Rel) = {(InDiv, idRel)}

δ(InDiv,class,Rel) = {(InDiv, ClassRel)}

δ(InDiv,style,Rel) = {(InDiv, StyleRel)}

δ(InDiv,<script,Div) = {(InDiv, BScriptDiv)}

δ(InDiv,>,BScript) = {(InScript, CScriptBScript)}

δ(InDiv,src,BScript) = {(InDiv, SourceBScript)}

δ(InDiv,=,Source) = {(InDiv, =SourceSource)}

δ(InDiv,"str",=Source) = {(InDiv, STR=Source)}

δ(InDiv,e,STR) = {(InDiv, e)}

δ(InDiv,e,=Source) = {(InDiv, e)}

δ(InDiv,e,Source) = {(InDiv, e)}

δ(InDiv,id,BScript) = {(InDiv, idBScript)}

δ(InDiv,class,BScript) = {(InDiv, ClassBScript)}

δ(InDiv,style,BScript) = {(InDiv, StyleBScript)}

z. Current State : F

δ(F,<!--,Z) = {(F, COMZ)}

δ(F,-->,COM) = {(F, e)}

1.6. Start State (q0)

q0 atau yang biasanya disebut starting state merupakan start awal dari sebuah PDA . Pada PDA ini, q0 adalah S, yang dimana input symbol awal yang diterima adalah <html.

1.7. Stack Symbol (Z0)

Z0 atau yang biasanya disebut initial stack symbol merupakan simbol yang digunakan pertama kali untuk stack PDA. Pada PDA kali ini, kami menggunakan simbol Z sebagai inisialisasi awal dari stack PDA kami.

1.8. Accepting States (F)

PDA ini merupakan tipe PDA *empty stack acceptance* sehingga hasil proses PDA dari sebuah string akan benar jika isi dari stack pada akhir telah kosong. Sebagai accepting states (F) , kami menggunakan state F. Pada state ini, jika inputan dalam stack kosong , maka inputan dinyatakan benar.

Bab III

Implementasi dan Pengujian

1.1. Spesifikasi Teknis Program

1.1.1. Struktur Data

STRUKTUR DATA PDA

Definisi type value dari key

```
pda = {
   ‘all_states’ : list of char,
   ‘input_symbols’ : list of string,
   ‘stack_symbols’ : list of string,
   ‘start_state’ : string,
   ‘start_stack_symbol’ : string,
   ‘final_state’ : list of string,
   ‘pda_type’ : char,
   ‘transitions’:dict,
}
```

Definisi type value ‘transition’ / object

```
transitions : {
   (currentState : string , input_symbol : string , pop_stack :
string ) : (goToState : string , push_stack : list of string)
}
```

1.1.2. Fungsi dan Prosedur

Program terbagi menjadi 3 file Python yaitu main.py dan parserPDA.py dan states.py . Di dalam file html berisi program utama yang akan menerima argumen dari inputan user pada terminal. Setelah mendapatkan nama file pda dalam bentuk txt dan file html yang akan dicek, akan melakukan pemanggilan beberapa fungsi dari file

parsePDA.py untuk meproses file html tersebut. Dalam file parsePDA.py, berisi fungsi parseHTML, parse, validate_tokens. Function parseHTML untuk memparse konten dari html menjadi token-token akan menjadi input dari PDA. Function parse untuk memparse PDA dari file pda.txt , kemudian fungsi validate_tokens untuk memvalidasi token dari html apakah accepted atau rejected.

File : parsePDA.py
def parseHTML(html_file : string) → list , list Fungsi ini akan menerima path dari nama file HTML yang ingin di cek kebenaranya. Kemudian file HTML ini akan di parse ke dalam list token sesuai dengan ketentuan yang berlaku. Setelah selesai memproses seluruh konten dari html, fungsi ini akan mengembalikan token-token hasil parse dalam bentuk list dan list yang berisi setiap line pada html. **def parse(file_path : string) → dict** Fungsi ini akan menerima file_path dari nama file txt yang berisi pda yang telah didefinisikan, lalu kemudian memproses isi dalam file tersebut ke dalam dictionary yang merupakan struktur data PDA. Kemudian setelah selesai fungsi ini mengembalikan data dictionary tersebut. **def validate_tokens(tokens : list , PDA : dict) → boolean, string , string , string, set** Fungsi ini menerima list token dan dictionary PDA , kemudian melakukan pemrosesan terhadap token tersebut dengan konsep PDA menggunakan stack. Pada akhir pemrosesan fungsi ini akan mengembalikan kebenaran dari tokens tersebut dalam bentuk boolean, input token terakhir dalam

bentuk string, nomor baris token tersebut pada html dalam bentuk string, dan terakhir ekspektasi input token apabila token tidak valid.

def output_to_terminal(validate:boolean, inputError:string , lineNumber : int , stack_symbol : string , expected : set , PDA : dict)
Fungsi ini menerima argumen berupa validasi , inputError , line Number , stack_symbol, expected , dan PDA . Kemudian fungsi ini memproses keluaran di terminal apakah file HTML tersebut ACCEPTED atau REJECTED.

File : main.py

```
Token, lineHTML = parseHTML(html_file)
PDA = parse(text_file)
validate_tokens(token_PDA)
output_to_terminal(validate,inputError,lineNumber,stack_symbol,expected,PDA)
```

Pertama, akan terlebih memparse HTML ke dalam bentuk list token, kemudian memparse file txt ke PDA. Setelah itu , akan memvalidasi token yang diperoleh berdasarkan PDA hasil parse. Kemudian hasil dari validasi akan di tampilkan di terminal.

File : states.py

Class Transition

def __init__def __init__(self, initial_state, input_symbol, stack_top, final_state, push_symbol)

Mengawali jalannya fungsi

def read_pda_file(file_path:strings)->Dict , array of strings

Membaca file PDA txt sebelum diproses ke fungsi lain

def find_matching_transitions(transitions:Dict,array of strings, initial_state:string, final_state:string)->array of strings

Menggabungkan semua PDA dalam satu baris dengan input initial state dan final state yang sesuai dengan input dan mengembalikan array baru dengan kecocokan initial state dan final state.

def print_all_unique_states(transitions:array of strings)

Memberikan output semua states , baik initial maupun final state namun dengan memastikan tidak ada yang berulang

def main()

Menjalankan semua program diatas, dengan memberikan tampilan hasil matching berupa hanya alphabet, stack alphabet dan push to stack symbol dari hasil matching tersebut.

1.1.3. Antarmuka

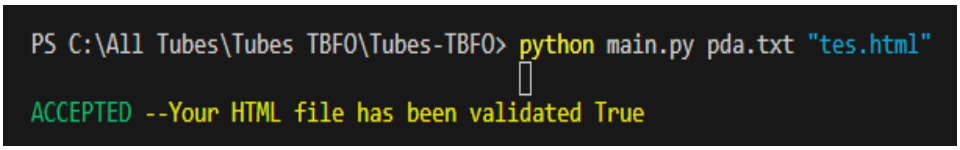
Interaksi antara program dengan pengguna dapat dijalankan dengan menggunakan program berikut:

```
> python main.py pda.txt "inputAcc.html"
```

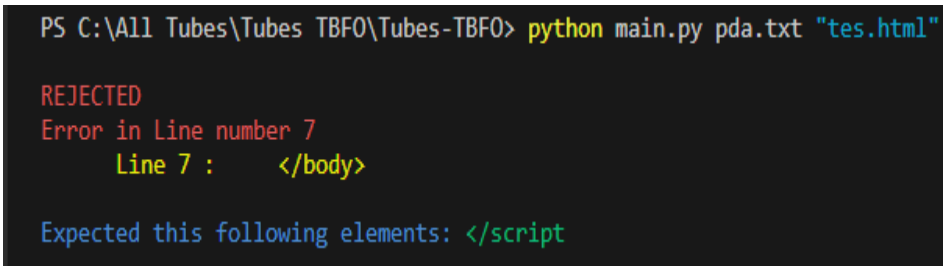
pda.txt merupakan file hasil konfigurasi state PDA yang telah dibuat dan "InputAcc.html" dapat diganti dengan file html yang ingin di cek kebenarannya.

Terdapat 2 jenis keluaran dari program ini yakni:

1. **Accepted,** yang ditampilkan pada CLI dengan luaran seperti berikut.



2. **Syntax Error,** yang akan ditampilkan saat ada error dalam file HTML yang diinput . Pesan Error akan berisi error di line apa, dan ekspektasi yang seharusnya agar file HTML tersebut benar.



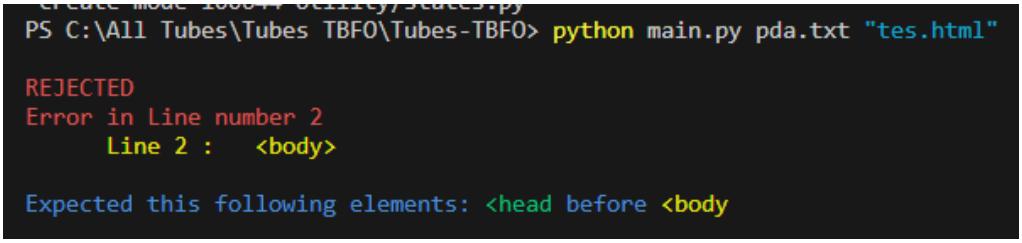
1.2. Uji Kasus

1.2.1. Kasus 1

```
<html>
  <body>
    <h1>Hello, World!</h1>
    <p>This is a simple webpage.</p>
  </body>
  <head>
```

```
    <title>Simple Webpage</title>
  </head>
</html>
```

Harapan luaran program yang diharapkan adalah program akan mengeluarkan pesan **REJECTED** saat mengecek file HTML yang diinput, karena dalam file html tersebut tag body muncul duluan sebelum tag head. Berikut merupakan tampilan luaran program.



1.2.2. Kasus 2

```
<hmif>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <p>This is a simple webpage.</p>
  </body>
</hmif>
```

Harapan luaran program yang diharapkan adalah bahwa program akan mengeluarkan pesan **REJECTED** saat mengecek file HTML yang diinput, karena pada syntax html salah penulisan menjadi hmif. Berikut adalah luaran program.

```
PS C:\All Tubes\Tubes TBFO\Tubes-TBFO> python main.py pda.txt "tes.html"

REJECTED
Error in Line number 1
      Line 1 : <hmif>

Invalid syntax at <hmif
```

1.2.3. Kasus 3

```
<html>
  <body>
    <h1>Hello, World!</h1>
    <p>This is a simple webpage.</p>
  </body>
</html>
```

Harapan luaran program yang diharapkan adalah program mengeluarkan pesan **REJECTED** saat membaca file HTML , karena dalam file html tersebut tidak terdapat tag head.Berikut adalah luaran program.

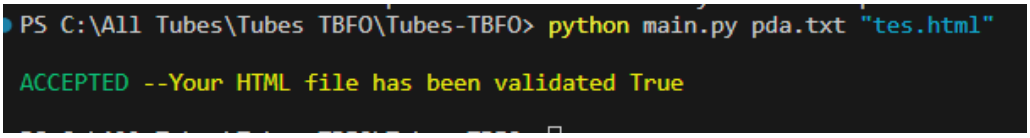
```
PS C:\All Tubes\Tubes TBFO\Tubes-TBFO> python main.py pda.txt "tes.html"

REJECTED
Error in Line number 2
      Line 2 :   <body>

Expected this following elements: <head before <body
```

1.2.4. Kasus 4

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    <img src=”./welcome.jpeg”>
    <p>This is a <em>simple</em> webpage.</p>
```

```
    <div id=”footer” class=”footer”> This is the end of the
page </div>
  </body>
</html>
```

Harapan luaran program yang diharapkan adalah program mengeluarkan pesan **ACCEPTED** saat membaca file HTML , karena file html tersebut sudah benar.Berikut adalah luaran program.



1.2.5. **Kasus 5**

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <!-- Bagian utama web -->
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    <hr>
    <img src="./welcome.jpeg" alt="Welcome Banner">
    <p>This is a <em>simple</em> webpage.</p>

    <!-- Custom element -->
    <div id=”footer” class=”footer”> This is the end of the
page </div>
  </body>
</html>
```

Harapan luaran program yang diharapkan adalah program mengeluarkan pesan **ACCEPTED** saat membaca file HTML , karena dalam file html tersebut sudah benar.Berikut adalah luaran program.

```
PS C:\All Tubes\Tubes TBFO\Tubes-TBFO> python main.py pda.txt "tes.html"

ACCEPTED --Your HTML file has been validated True
```

1.2.6. Kasus 6

```
<html>
  <head>
    <title>Simple Webpage</title>
  </head>
  <body>
    <!-- Bagian utama web -->
    <h1>Hello, World!</h1>
    <h2>Welcome to my page</h2>
    <img alt="Welcome Banner">
    <p>This is a <em>simple</em> webpage.</p>

    <!-- Custom element -->
    <div id="footer" class="footer"> This is the end of the
page </div>
  </body>
</html>
```

Harapan luaran program yang diharapkan adalah program mengeluarkan pesan **REJECTED** saat membaca file HTML , karena dalam file html tersebut tag img tidak terdapat atribut src.Berikut adalah luaran program.

```
PS C:\All Tubes\Tubes TBFO\Tubes-TBFO> python main.py pda.txt "tes.html"

REJECTED
Error in Line number 9
      Line 9 :     <img alt="Welcome Banner">

Expected this following elements: src before >
```

1.2.7. Kasus 7

```
<html>
```

```
<head>
  <title>Simple Webpage</title>

</head>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php" method="POST">
  <h5 class="label">First name:</h5><br>
  <input type="text" id="fname"><br>
  <h5 class="label">Last name:</h5><br>
  <input type="text" id="lname"><br><br>
  <button type="submit">Submit</button>
</form>

<p>If you click the "Submit" button, the form-data will be
sent to a page called "/action_page.php".</p>

</body>
</html>
```

Harapan luaran program yang diharapkan adalah program mengeluarkan pesan **ACCEPTED** saat membaca file HTML , karena dalam file html tersebut sudah valid.Berikut adalah luaran program.

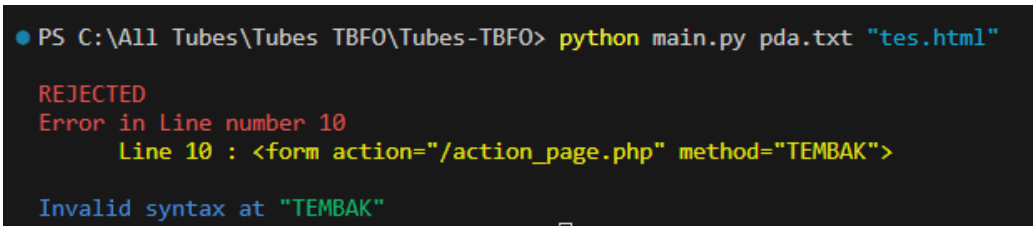
```
PS C:\All Tubes\Tubes TBFO\Tubes-TBFO> python main.py pda.txt "tes.html"
ACCEPTED --Your HTML file has been validated True
```

1.2.8. **Kasus 8**

```
<html>
<head>
  <title>Simple Webpage</title>
```

```
</head>
<body>

<h2>HTML Forms</h2>

<form action="/action_page.php" method="TEMBAK">
  <div id="label">First name:</div><br>
  <input type="text" id="fname"><br>
  <div id="label">Last name:</div><br>
  <input type="text" id="lname"><br><br>
  <button type="submit">Submit</button>
</form>

<p>If you click the "Submit" button, the form-data will be
sent to a page called "/action_page.php".</p>

</body>
</html>
```

Harapan luaran program yang diharapkan adalah program mengeluarkan pesan **REJECTED** saat membaca file HTML , karena dalam file html tersebut atribut method pada tag form tidak valid.Berikut adalah luaran program.

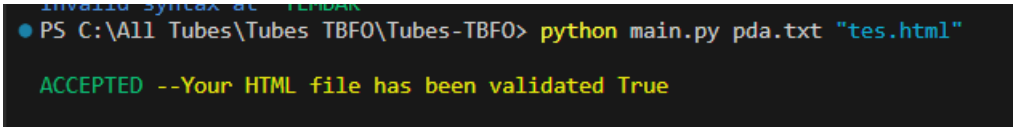


1.2.9. Kasus 9

```
<html>
<head>
  <title>Simple Webpage</title>
  <script>
```

```
    document.getElementById("demo").innerHTML = "Hello
JavaScript!";
  </script>
</head>
<body>

<h1>The script element</h1>

<p id="demo"></p>

</body>
</html>
```

Harapan luaran program yang diharapkan adalah program mengeluarkan pesan **ACCEPTED** saat membaca file HTML , karena dalam file html tersebut sudah benar.Berikut adalah luaran program.



1.2.10. Kasus 10

```
<html>
<head>
  <title>Simple Webpage</title>
  <script>
    document.getElementById("demo").innerHTML = "Hello
JavaScript!";
  </script>
</head>
<body>

<h1>The script element</h1>

<p id="demo">
```

```
</body>
</html>
```

Harapan luaran program yang diharapkan adalah program mengeluarkan pesan **REJECTED** saat membaca file HTML , karena dalam file html tersebut tidak terdapat tag penutup untuk </p.Berikut adalah luaran program.

```
PS C:\All Tubes\Tubes TBFO\Tubes-TBFO> python main.py pda.txt "tes.html"

REJECTED
Error in Line number 14
      Line 14 : </body>

Expected this following elements: </p before </body
```

1.2.11. **Kasus 11**

```
<html>
    <head>
        <title>Simple Webpage</title>
        <script>
            document.getElementById("demo").innerHTML =
"Hello JavaScript!";
        </script>
    </head>
    <body>
        <h1>The script element</h1>
        <a>Not going anywhere</a><br>
        <a href="https://www.google.co.id/">Might send you
somewhere</a>

        <p id="demo"></p>
    </body>
</html>
```

Harapan luaran program yang diharapkan adalah program mengeluarkan pesan **ACCEPTED** saat membaca file HTML , karena dalam file html tersebut sudah valid.Berikut adalah luaran program.

```
PS C:\All Tubes\Tubes TBFO\Tubes-TBFO> python main.py pda.txt "tes.html"
ACCEPTED --Your HTML file has been validated True
```

Bab IV

Tautan dan Pembagian Tugas

1.1. Link Repository GitHub

https://github.com/Benardo07/Tubes-TBFO-Kelompok-15-Jimmy-88

1.2. Link Diagram State

https://drive.google.com/file/d/1RS_04R1selu7c1R1SKmv52UT4Ba0awut/view

https://app.diagrams.net/#G1RS_04R1selu7c1R1SKmv52UT4Ba0awut

1.3. Pembagian Tugas

No	Nama	Tugas
1.	Zachary Samuel T	Membantu merancang PDA
2.	Wilson Yusda	Laporan, Membantu merancang PDA, Diagram PDA, states.py (pencocok initial dan final state)
3.	Benardo	PDA , parser html , parser PDA ke struktur data dictionary, merancang seluruh transition function pada PDA, laporan

Bab V

Daftar Pustaka

HTML TUTORIAL :
https://www.w3schools.com/html/default.asp

PDA REFERENCE:
https://www.geeksforgeeks.org/introduction-of-pushdown-automata/