

Trabalho teórico-prático

2024/1

Dennis Pereira Krigger, 00288746
Leonardo Azzi Martins, 00323721

Descrição do cenário

Uma Uniform Resource Locator, ou URL, se trata de uma referência a um recurso que especifica sua localização em uma rede como a internet. Toda URL é um tipo de Uniform Resource Identifier (URI), que representa uma sequência única de caracteres que identificam recursos genéricos, que vão desde as páginas da web até endereços de e-mail. URLs típicas, como <https://www.inf.ufrgs.br/site/>, são compostas pela sintaxe de uma URI genérica, tendo um *scheme* (https), um *host* (www.inf.ufrgs.br) e um *path* (site). Existem outros elementos que podem compor a sintaxe, como *userinfo*, *port*, *query* e *fragment*.

URI = scheme ":" ["/" "/" [userinfo "@"] host [":" port]] path ["?" query] ["#" fragment]

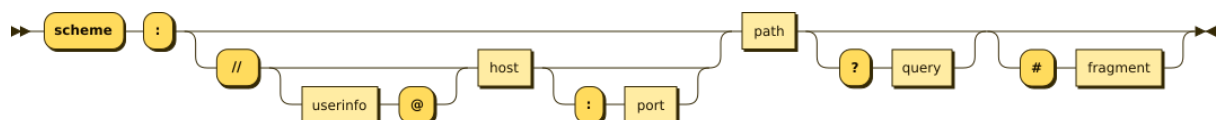


Diagrama da sintaxe genérica de uma URI

O *scheme* e o *path* são sempre definidos para todas as palavras válidas da sintaxe. Os outros componentes têm alguns delimitadores associados.

- **scheme:** componente acompanhado de um delimitador dois-pontos (:). Alguns *schemas* populares são http, https, ftp, mailto, file, data, entre outros.
- **authority:** componente opcional, precedido por duas barras (//) e composto pelos sub-componentes *userinfo* e *host*.

- **userinfo:** opcional, seguido pelo símbolo arroba (@), consiste em um nome de usuário e uma senha opcional.
- **host:** consiste de um nome registrado, que pode ser um *hostname* ou endereço de IP.
- **port:** opcional, precedido pelo símbolo dois-pontos (:), consiste de um conjunto dígitos que corresponde ao endpoint para um serviço específico.
- **path:** realiza o mapeamento para um caminho no sistema de arquivos, através de referências lógicas.
- **query:** opcional, precedido do símbolo interrogação (?), consiste em uma string de consulta para dados não-hierárquicos.
- **fragment:** opcional, precedido do símbolo *hash* (#), consiste de um identificador que direciona para um recurso secundário, como seções de um *heading* em HTML.

Existem outros símbolos e sub-componentes que, para fins de viabilidade e simplificação, não foram considerados no escopo deste trabalho.

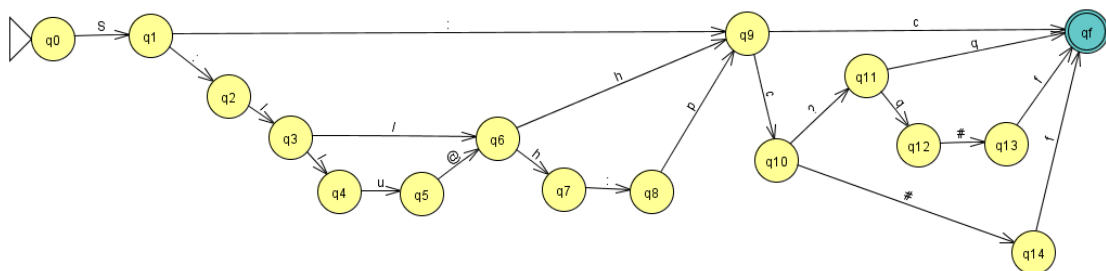
Definição da linguagem

$$L_c = \{S: (/(u@)^x h(:p)^y)^n c (?q)^i (\#f)^j \mid 0 < n, x, y, i, j \leq 1\}$$

$$\Sigma = \{S, :, /, h, u, p, c, q, f, ?, @, \#\}$$

Símbolo de Σ	Descrição do símbolo
:	Símbolo ASCII dois-pontos, sucede um <i>schema</i> e precede um <i>port</i>
/	Símbolo ASCII de barra, que precede o componente <i>authority</i>
<i>S</i>	Schema, que pode representar protocolos para páginas web HTTP ou HTTPS, protocolo de transferência de arquivos FTP, protocolo para e-mail mailto, entre outros
<i>h</i>	Host, consiste de um nome registrado, que pode ser um hostname ou endereço de IP
<i>u</i>	Userinfo, consiste em um nome de usuário e uma senha opcional
<i>p</i>	Port, consiste de um conjunto de dígitos que corresponde ao endpoint para um serviço específico
<i>c</i>	Path (caminho), realiza o mapeamento para um caminho no sistema de arquivos, através de referências lógicas
<i>q</i>	Query, consiste em uma string de consulta para dados não-hierárquicos
<i>f</i>	Fragment, consiste de um identificador que direciona para um recurso secundário, como seções de um heading em HTML
#	Símbolo ASCII <i>hash</i> que precede o <i>fragment</i>
@	Símbolo ASCII arroba que sucede o <i>userinfo</i>
?	Símbolo ASCII interrogação que precede a <i>query</i>

AFN que reconhece a linguagem



Regras de formação das palavras

Exemplos de palavras em $ACEITA(M_c)$:

- $S: c$
- $S: //hc$
- $S: //h: pc$
- $S: //u@h: pc$
- $S: //u@h: pc? q\#f$

Exemplos de palavras em $REJEITA(M_c)$:

- $SS: /c$
- $S: /c$
- $S: //ch: p$
- $S: //u@h: p\#f? q$
- $S: //u@h: p$