التعابير النمطية

Regular Expressions

مؤيد السعدي

التعابير النمطية

التعابير النمطية هي المكون الأساسي لنموذج رياضي للغات يسمى اللغات النمطية المعرفة على أبجدية ما.
ويرتبط بها الكثير من مجالات البحث الأكاديمية وله تطبيقات عملية عديدة منها:

- فحص المطابقة (سلامة المدخلات)
- البحث عن ابرة في كومة نص
- التقطيع والإعراب والفصل
- الاستبدال والتسوية normalization

تطبيقات محتملة

- البحث عن الملف الذي يحتوي تعريف دالة ما
- البحث عن كل الملفات التي تستدعي دالة ما
- البحث عن دالة لا تعرف اسمها بالضبط مثلا Map ثم شيء ثم View أو Show
- تبديل المعامل الثالث والرابع في دالة ما
- استخراج كل عناوين البريد حيث الحساب نفس النطاق مثلا ahmad@ahmad.com
- استخراج كل الروابط
- تحويل كل www. إلى روابط

تطبيقات محتملة

- تحويل كل الروابط إلى nofollow
- تحويل نص HTML مرقوم إلى نص صرف
- ترقيم نص صرف إلى نص مرقوم بأنماط معينة
- استخراج الكلمات التي تحتوي حروف مكررة متتالية في نص
- تسوية الهمزات قبل فهرسة النص
- إزالة الحركات
- استبدال التواريخ من هيئة إلى أخرى مثل YYYY-MM-DD

تطبيقات محتملة

- تحويل وسم strong إلى وسم b في HTML
- كلمة تبدأ بحرف s وتحتوي حرفين مكررين مرتين مثل seedless
- كلمة تحتوي مقطع من 3 حروف مكرر مرتين مثل abracadabra
- تنقية ملف HTML من وسوم javascript

تطبيقات محتملة

- قلب الاسم الأول والأخير مع وضع فاصلة
  Adams, John
- سبر تقارير خادم Apache لحصر عنوان IP الأكثر
  - من حيث عدد الضربات
  - من حيث كمية التنزيل
- سبر التقرير بحثا عن أكثر الصفحات شيوعا أو أكثر الأنواع شيوعا أو أكثر المواقع إحالة

تطبيقات مستحيلة دون عمل خارجي

- استخراج البريد الذي طول الاسم فيه نفس طول النطاق
- قلب الحروف الكبيرة والصغيرة
- جمع الأرقام
- استبدال الكلمات الطويلة بأول حرف وآخر حرف وبينهما عدد الحروف مثل L10n

أنواع التعابير النمطية

هناك أنواع أو لهجات من التعابير النمطية منها القياسية ومنها الممتدة

- العيارية الأساسية POSIX BRE
- العيارية الممتدة POSIX ERE
- PCRE في لغة Perl وغيرها
  - php
  - Javascript
  - apache config
- Python

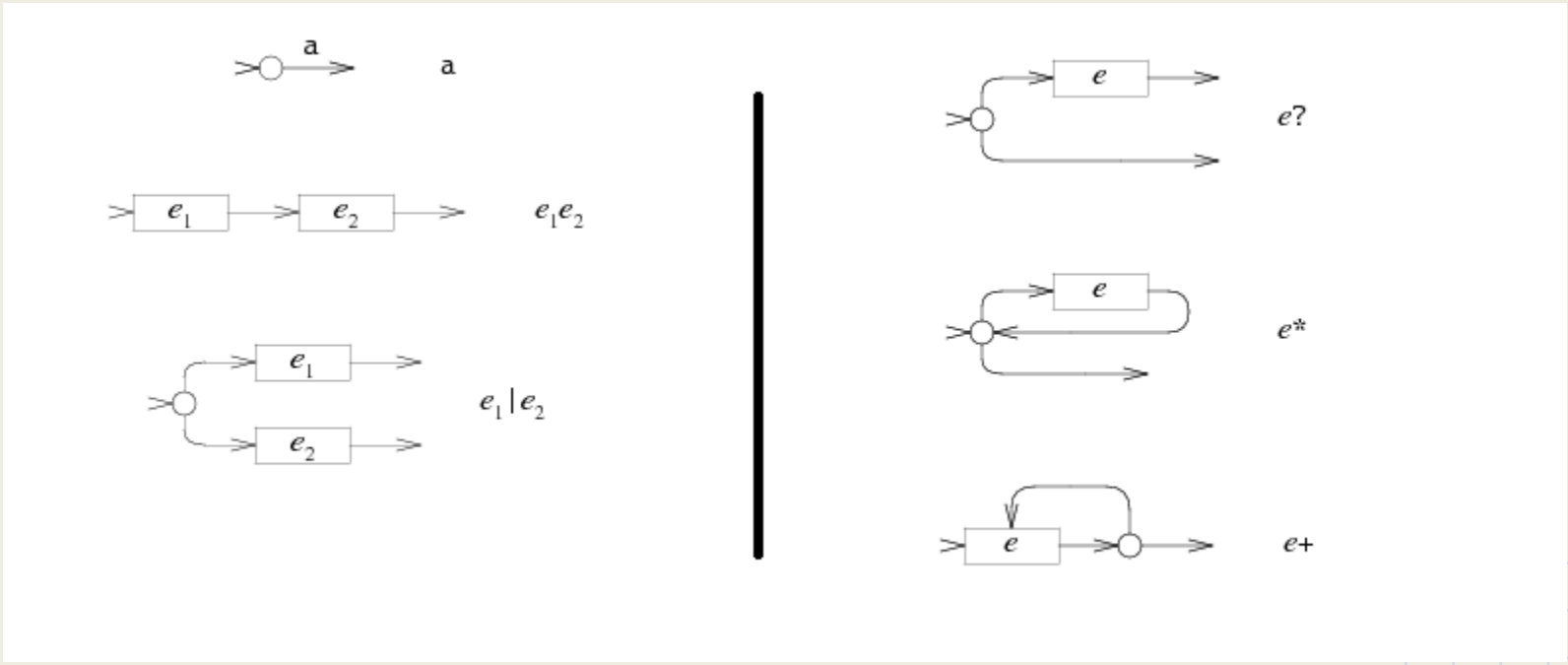
التعابير الأساسية

هذا مجرد تمهيد للتعابير الأساسية BRE

- المحرف العادي يعبر عن نفسه مثل a
- النجمة ترمز لتكرار التعبير السابق أي عدد من المرات
- النقطة أي محرف واحد
- | للتخيير بين التعبير السابق والتالي

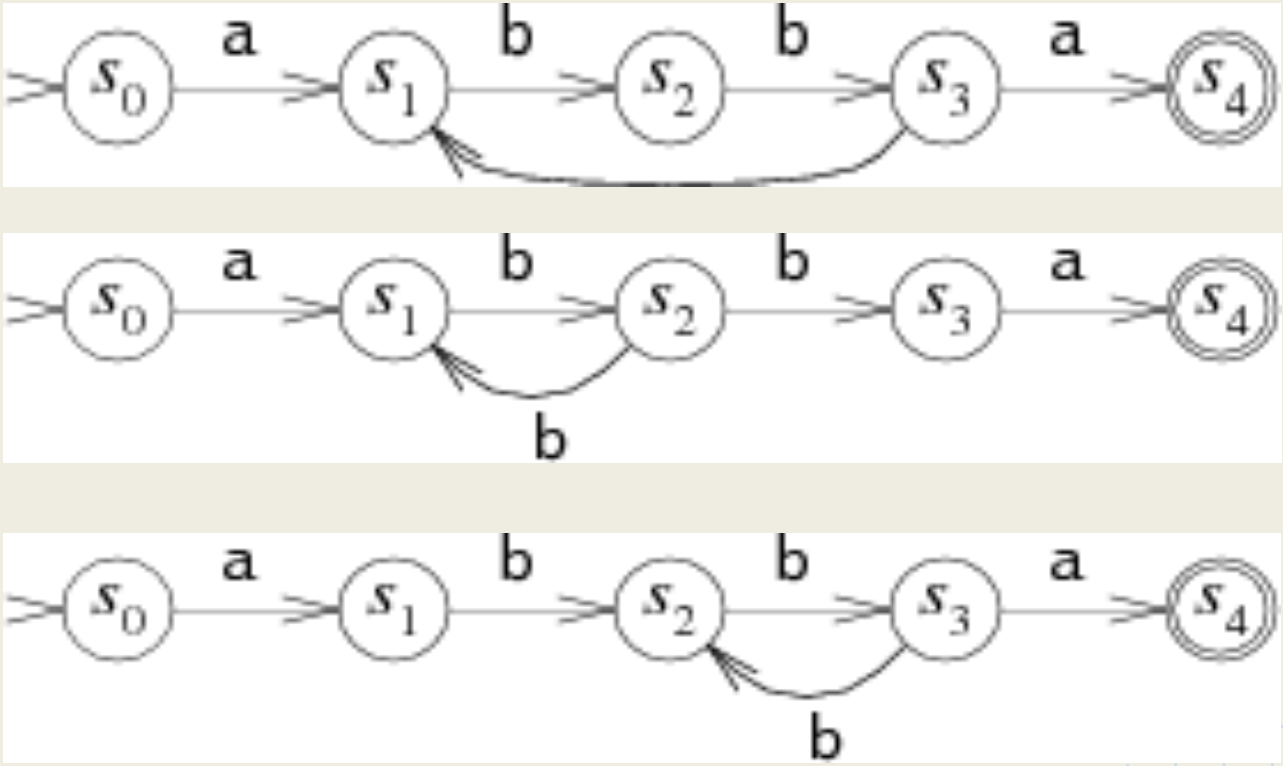
التمثيل على شكل NFA و DFA

يمكن تمثل أي BRE على شكل NFA وهو ليس فريد



التمثيل على شكل NFA و DFA

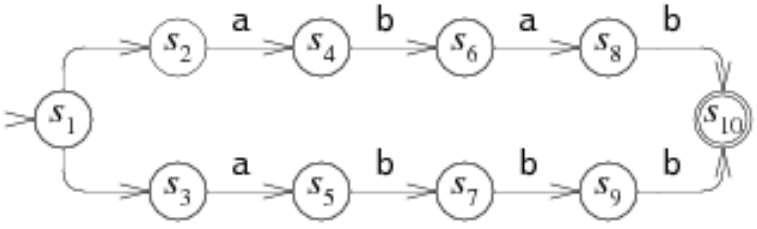
التمثيل ليس فريد مثلا a(bb)*a يمكن أن يكون



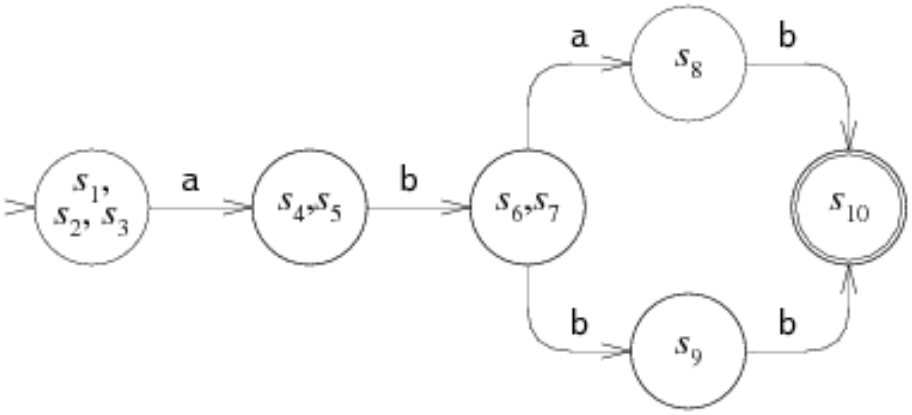
التمثيل على شكل NFA و DFA

إذا حولنا NFA إلى DFA يمكن تسهيل تنفيذها

abab | abbb, with state numbers added:



abab | abbb



abab | abbb

التصنيف و التنفيذ

قبل أن يتم مطابقة التعبير النمطي يجب أن يتم تصنيفه compile وإن ضمنيا لهذا تسمى عملية فحص المطابقة تنفيذا execute لما تم تصنيف سابقا.

لأسباب تاريخية كان يتم تحويل النمط إلى لغة الآلة وكأنها JIT لكن حاليا غالبا ما يعني التصنيف مجرد إنشاء الكائن واستهلاله بالتركيب المناسب NFA أو DFA. ويكن تنفيذ نفس التعبير على أكثر من نص بتصنيف واحد

محددات التكرار

يمكن تحديد تكرار التعبير النمطي السابق بأي مما يلي

- ? تعبير اختياري: مكرر صفر أو واحد
- * مكرر صفر أو أكثر
- + تعبير إلزامي قابل للتكرار: مكرر 1 أو أكثر
- {n} مكرر n مرة
- {,n} مكرر n أو أكثر
- {n,m} مكرر من n إلى m من المرات شاملا الحدود

مكان الرسو anchor

يمكن اشتراط مكان المطابقة (لا تستهلك من المدخلات)

- ^ في البداية
- $ في النهاية
- \b حد الكلمة boundary (سنأتي على معنى كلمة)
- \B في وسط الكلمة
- <> تحصر بداخلها كلمة منفردة

ملاحظات:

- ^$ تعني سطر خال

التخيير عبر علامة |

يمكن التخيير بين التعبير السابق والتالي مع مراعاة أن التكرار أولى من التتابع والذي هو أولى من التخيير

- ab|c$
  - (ab)|(c$)
- ab*
  - a(b*)

الأقواس

الأقواس تجمع التعابير معا بما يتعدى الأولوية كما أنها تجمع ما تتطابق داخلها كعلامة مرجعية back reference يمكن الإشارة لها لاحقا عبر رقم القوس مسبوق بعلامة \ أو $ حسب اللغة.

لمطابقة بريد إلكتروني اسمه بنفس اسم النطاق مثل ali@ali.com نستعمل

^(.*)@\1\.com$

المجموعات sets

يمكن اشتراط ورود محرف واحد من بين مجموعة

- [0123456789ABCDEF]
  - منزلة ستعشرية
- [0-9A-F]
  - نفس سابقتها

أو عدم وروده من بينها

- [^0-9]
  - أي محرف واحد ليس منزلة عشرية

اختصارات المجموعات

هناك بعض الاختصارات الشائعة منها

- \d \D
  - [0-9] [^0-9]
- \w \W
  - [0-9A-Za-z_] [^0-9A-Za-z_]
- \s \S
  - [\t\r\n\v] [^ \t\r\n\v]
- mix like eg. [^\w\s] [+\-\.\d]

محددات الأطوار

هناك عدد من أطوار العمل يمكن التحكم فيها عبر محددات أثناء التصنيف modifiers منها

- i تجاهل التمييز بين الحروف الكبيرة والصغيرة
- m تفعيل طور تعدد الأسطر مما يؤثر على ^$
- g تعني global وتطابق أكثر من مرة
- s تعني أن . تشمل علامة سطر جديد
- u تعني أن \w تشمل حروف لغات أخرى

معضلة التخطي

لاحظ التخطي في لغة سي++ لتمثيل مسار ملفات

```
cout << “C:\\Windows\\System\\”
```

نعاني من شيء مشابه في التعابير النمطية حيث علينا إيصال ما نريد للدالة متخطين المعاني الخاصة لبعض المحارف مثلا \1 في تعني المحرف الذي رمزه 1 في جدول ASCII وليس المجموعة المرجعية الأولى.

معضلة التخطي

في لغة بيرل لدينا للتخطي

\Q...\E

في لغة PHP هناك دالة addslashes

في لغة python نضع r قبل التعبير

أمثلة عملية

افتح ملف HTML المرفق واختر Replace ثم اكتب

- Pattern: ^(\S+), (\S+)$
- Replace: $2 $1
- Input Text:
  - Smith, John.
  - Omar, Khalil
  - Ali, Ahmad.

أمثلة عملية

افتح ملف HTML المرفق واختر Replace ثم اكتب

- Pattern: hat
- Replace: ###
- Input Text:
  - That is not my hat

أمثلة عملية

افتح ملف HTML المرفق وافتح أدوات التطوير واكتب

```
id('in0').value.replace(/^(\S+), (\S+)$/mig, id('rep0').value);
```

وجرب

```
id('in0').value.replace(new RegExp("^(\\S+), (\\S+)$", "mig"), id('rep0').value);
```

أمثلة عملية

ماذا تفعل الأوامر التالية:

- `egrep -i '^(\w).*\1$' /usr/share/dict/words | head`
- `egrep -i '^s(\w)\1.*(\w)\2' /usr/share/dict/words | head`
- `egrep -i '^(\w{3,}).*\1.*' /usr/share/dict/words | head`
- `perl -Mutf8 -CASD -lwpe 's![اأإ]!ا!gim;' ar.txt`

في لغة بيرل عند التعامل مع اللغة العربية ومحارف UTF8 نستخدم CASD ونفعل الوحدة utf8 التي تكافئ

use utf8

جشع التعابير النمطية

التعابير النمطية جشعة بطبعها فهي تبحث عن إلتهام أكبر قدر من المدخلات عند المطابقة فهي تبحث عن أطول نص وهذا شيء جيد في الغالب لكنه قد يطعنك في ظهرك إن لم تكن تتوقع.

معضلة جشع التعابير النمطية

لنقم بوضع وسم مكان وسم في ملف HTML يحتوي ما يلي

abc def ghi jkl mno

مستخدمين التعبير (.*) ونلاحظ أنه يطابق المنطقة الحمراء

abc **def ghi jkl** mno

وبعد الاستبدال سيكون الناتج

abc def**** ghi ****jkl mno

حل إلتفافي

لنقم بوضع وسم مكان وسم في ملف HTML يحتوي ما يلي

abc def ghi jkl mno

مستخدمين التعبير ([^<]*) وسينجح لأننا استثنينا بداية الوسم

وبعد الاستبدال سيكون الناتج

abc def**** ghi ****jkl mno

التعابير القنوعة

إذا وضعت ? بعد * أو + يتحول التعبير إلى تعبير قنوع يبحث عن أقصر مطابقة.

لنقم بوضع وسم مكان وسم في ملف HTML يحتوي ما يلي

abc def ghi jkl mno

مستخدمين التعبير (.*?)

وبعد الاستبدال سيكون الناتج

abc def ghi jkl mno

تطبيق عملي: تنظيف HTML

نريد تنظيم ملف HTML من سكربتات Javascript
ادرس التعبير النمطي التالي:

<script[^>]*>.*?</script>

تلقائيا علامة . لا تشمل علامة سطر الجديد. المحدد s
يجعلها كذلك في لغة بيرل وبايثون

```
perl -wlp0e 's!<script[^>]*>.*?
</script>!!gims;' myfile.html
```

تطبيق عملي: تنظيف HTML

نريد تنظيم ملف HTML من سكربتات Javascript
ادرس التعبير النمطي التالي:

```
<script[^>]*>(.|[\r\n])*?</script>
```

تجاوز نقاط القصور

لنقم بما قلنا أنه مستحيل. نطبق على المدخلات التالية.

- ab
- abc
- Internationalization
- Localization

```
id('in0').value.replace(/(\w)(\w\w+)(\w)
/gim, function(m,g1,g2,g3){return g1+
(g2.length)+g3;});
```

عندما لا تفي RegEx بما نحتاجه نعود لدوال اللغة الحاضنة.

تجاوز نقاط القصور

لنقم بما قلنا أنه مستحيل. نطبق على المدخلات التالية.

- ab
- abc
- Internationalization
- Localization

```
id('in0').value.replace(/(\w)(\w\w+)(\w)
/gim, function(m,g1,g2,g3){return g1+
(g2.length)+g3;});
```

عندما لا تفي RegEx بما نحتاجه نعود لدوال اللغة الحاضنة.

تجاوز نقاط القصور

جرب الأمر التالي:

```
echo "internationalization" |
  perl -lwpe 's!^(\w)(\w\w+)(?{ $I =
length($^N) })(\w)!$1$I$3!g;'
```

تعابير تنظر دون أن تأكل

أحيانا تريد أن تتطلع من حولك دون أن تلتهم ما تتطلع إليه من مدخلات.

- التطلع للأمام إيجابا
- positive look-ahead assertion
- /\w+(?=\t)/
- التطلع للأمام سلبا
- s/foo(?!bar)/###/g

تعابير تنظر دون أن تأكل

- التطلع للخلف إيجابا
- positive look-behind assertion
- s/(?<=foo)bar/###/g
- التطلع للخلف سلبا
- s/(?<!foo)bar/###/g

تمارين

ماذا يفعل الأمر التالي

```
echo -e "this banana-eating-monky is too big" | perl -lwne 'while(m!(\b\w{2,3}\b)!g){print $1;}'
```

شكرا لكم

للتواصل معي

alsadi@gmail.com
twitter: @muayyadalsadi
http://g0alkeeper.blogspot.com
https://github.com/muayyad-alsadi/