# تطبيقات واستنتاجات عن اساليب الاعراب

## Application and conclusions on some parsing Techniques

د .زكريا الصوفي
السيد كاظم زعيل الموسوي

### 1 – مقدمة

سبق وأن شرحنا في العدد الرابع من المجلة (1) المرحلة الأولى من مراحل المترجم (Compiler) وكذلك تبينا دور محلل المفردات (Lexical Analyser) وكيفية عمله ومدى أهميته في عمل المترجمات .

أما في هذا العدد فسوف نحاول أن نشرح المرحلة الثانية من مراحل المترجم وهي مرحلة الأعراب (Parsing) أو ال (Syntax) وذلك من خلال تعريف عمل المحلل النحوي ومناقشة أهم الطرق الشائعة في هذا المجال بعد ذلك وعلى ضوء التركيب الخاص للغة البرمجة العربية المستحدثة (ليث) تناقش كيفية أختيار طريقة الأعراب المناسبة لهذه اللغة ، حيث قد تم كتابة برنامج المحلل النحوي للغة وبلغة البرمجة العليا (فور تران) وتم تنفيذه بنجاح على حاسبة المركز القومي للحاسبات الألكترونية وكذلك على حاسبة الشركة العامة للمقاولات الأنشائية .

### 2 – تعريف عمل المحلل النحوي SYNTAX ANALYSER (2)

أن عملية فحص سلسلة الرموز (String of Tokens) الخارجية من مرحلة تحليل المفردات (Lexieal) لمعرفة تطابقها مع تركيب كلام اللغة الأصلي تسمى بعملية الأعراب (Parsing) أو ال (Syntax. Analysing).

هذا يعني أن أية جملة (نص) من جمل برنامج المصدر وبعد الأنتهاء من مرحلة تحليل مفرداتها تتم عليها عملية فحص وبواسطة خوارزمية أعراب معينة وذلك لمعرفة مطابقة تركيبها اللغوي مع تركيب اللغة والموصوف مسبقا أو عدم تطابقه .

**3 – طرق تصميم المحلل النحوي (طرق الأعراب)**

بصورة عامة يمكن تقسيم طرق الأعراب الى قسمين رئيسيين ، القسم الأول تسمى بالطرق غير الحتمية (Non-Deterministic) ، أما القسم الثاني فتسمى الحتمية (Deterministic) .

وتصنف طرق كل قسم الى نوعين في النوع الأول يبدأ المحلل عملية أعراب النص من جذر التفرع (ROOT*) ويعمل الى أن يجد الهدف (GOAL) هذا النوع من الطرق تسمى بطرق ال (TOP DOWN) . أما النوع الثاني فيبدأ المحلل من نهاية التفرع والذي يمثل الهدف (GOAL) ، ويعمل الا أن يجد جذر (ROOT) ذلك التفرع أو تحديد نوع الخطأ الموجود فيه ، ويسمى بطرق ال (BOTTOM-UP) ، ولكل من هذه الطرق محاسنها ومساوئها وتقرر حسب قواعد لغات البرمجة المستخدمة .

**3 . 1 الطرق غير الحتمية NON-DETERMINISTIC .**

في هذا النوع من الطرق يقوم المحلل بأفتراض (تخمين) قرارات أعراب معينة ثم الرجوع الى نقطة الأنطلاق (النقطة التي تم الأفتراض منها) في حالة عدم الوصول بواسطة هذه القرارات الى تقرير صحة النص الداخل أو عدم صحته .

تقسم طرق الأعراب غير الحتمية الى نوعين رئيسيين ، النوع الأول يسمى بطرق - TOP DOWN أما النوع الثاني فتسمى بطرق الـ BOTTOM-UP من الطرق الشائعة الأستخدام في كلا النوعين طريقة التعقب الرجعي والتي يمكن شرح عملها بالصورة التالية :-

**3 . 1 . 1 طريقة التعقب الرجعي (BACK - TRACKING) (3)**

وهي الطريقة المصممة من قبل (Fleyd 1964) والتي تستخدم أسلوب التعقب الرجعي (Back - Tracking) ولشرح عملها نستخدم المثال التالي :-

* – لتوضيح معنى GOAL, ROOT يرجى الرجوع الى الاشكال رقم (1 ، 2) من هذا البحث .

لو كانت لدينا القواعد التالية :-

$S \longrightarrow AX$

$A \longrightarrow V$

$A \longrightarrow B$

$B \longrightarrow VW$

حيث :-

B, A, S هي رموز غير نهائية Non Terminal

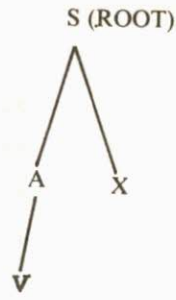
W, V, x هي رموز نهائية Terminal

S هو رمز الأنطلاق Starter Symbol (ROOT)

والمطلوب هو معرفة أمكانية توليد سلسلة الرموز الداخلة (VWX) من هذه القواعد أم لا ؟

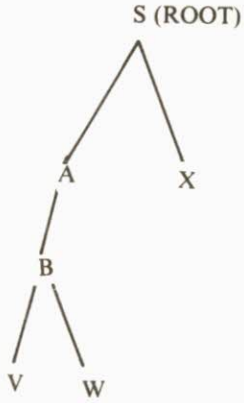
بما أن (s) هو رمز الأنطلاق فيجب علينا أولا تمييز هذا الرمز ، حيث نأخذ أول قاعدة تحتوي في جهة اليسار منها على الرمز (S) وفي المثال أعلاه تكون القاعدة الأولى ، حيث تعطينا السلسلة (AX) ولكن الرمز (A) هو رمز غير نهائي فيجب علينا أن نميزه ، ويتم ذلك بأخذ أول قاعدة تحتوي في جهة اليسار منها على الرمز (A) وهي القاعدة الثانية من القواعد أعلاه والتي تعطينا الرمز النهائي (V) وبذلك حصلنا على السلسلة (VX) (شكل رقم 1) ولكن هذه السلسلة لا تتطابق مع السلسلة الداخلة (VWX) وبذلك يجب علينا أن نرجع الى الرمز (A) لغرض تمييزه مرة أخرى حيث نترك القاعدة الثانية ونأخذ القاعدة التالية التي تحتوي في جهة اليسار منها على الرمز (A) وهي القاعدة الثالثة والتي تعطينا الرمز غير النهائي (B) وبذلك يتحتم علينا أولا تمييز الرمز (B) وذلك بأخذ أول قاعدة تحتوي في جهة اليسار على الرمز (B) وهي القاعدة الرابعة حيث تعطينا السلسلة (VW) وبذلك تم تمييز الرمزين (A, B) على التوالي وأيضا تم تمييز الرمز (S) وحصلنا على السلسلة (VWX) (شكل رقم 2) وبما أن هذه السلسلة تتطابق مع الرموز الداخلة (VWX) فأن عملية الأعراب قد أنتهت .

هذه الخوارزمية (ALGORITHIM) تسمى بال (الرجوع البطيء- Slow Back) ولكن توجد طريقة أخرى أكفأ منها تسمى بال (الرجوع السريع FAST BACK) والفرق بينهما هو أنه في هذه الطريقة يتم تمييز الهدف الثانوي لمرة واحدة فقط وعلى ضوء ذلك يتم تمييز الهدف الرئيسي .




GOALS هي W, V, X

، شكل رقم –1–




شكل رقم –2–

### 3 . 2 الطرق الحتمية DETERMINISTIC

في هذا النوع من الطرق يتخذ المحلل سلسلة من القرارات النهائية التي تقودنا وبصورة مباشرة لمعرفة النص الداخل فيما اذا كان موجود في اللغة أم لا دون الرجوع الى نقطة الانطلاق (No – Back Tracking) – وتقسم هذه الطرق الى الانواع التالية : –

### 3 . 2 . 1 طرق الـ BOTTOM – UP

هناك طرق عديدة من هذا النوع ، وسوف نشرح بصورة موجزة أهم الطرق الشائعة الاستعمال : –

#### 3 . 2 . 1 . 1 . طريقة الـ OPerator precedence parser (2)

هذه الطريقة تتخذ القرار حول صحة النص او عدم صحته استنادا الى اسبقية العوامل (Operaters) المستخدمة في اللغة ، اي كل عامل تعطي له اسبقية وهذه القيم تعتبر كسيطرة في عملية الاعراب .

القواعد التي تستخدم لها هذه الطريقة تسمى بالـ (Operater Grammer)والتي لا توجد فيها (Production) تحتوي في جهة اليمين منها على اثنين متتاليين من الرموز غير النهائية (4) – (Non terminal) .

### 3 . 2 . 1 . 2 طريقة Simple precedence parser (5)

هذه الطريقة تتخذ القرار حول صحة النص او عدم صحته استنادا الى درجة اسبقية العلاقات بين الرموز (SymBOLS) وفي هذا النوع تستخدم ثلاث علاقات للاسبقية وهي (<. ، ≐ ، .>) لفصل او عزل ، Handle في الـ (Right – Sentential form) التالي (∝BW) حيث انه اذا كانت B هي الـ Handle فأن العلاقة .> أو ≐ هي للفصل بين كل ازواج الرموز (Pairs of symbol) الموجودة في ∝ اما العلاقة .> فهي للفصل بين اخر رمز من ∝ واول رمز من B ، والعلاقة ≐ هي للفصل بين ازواج الرموز في الـ Handle نفسه ، والعلاقة <. هي للفصل بين اخر رمز في B واول رمز من W .

وتوجد طريقة اخرى ايضا تتخذ القرار استنادا الى درجة اسبقية العلاقات بين الرموز تسمى طريقة الـ (Weak precedence parser) (4) .

### 3 . 2 . 1 . 3 طريقة الـ Extended precedence parser (5)

هذه الطريقة تتخذ القرار حول صحة النص أو عدم صحته استنادا الى درجة اسبقية العلاقات بين سلاسل الرموز حيث ان عمل هذه الطريقة يشابه عمل الطريقة السابقة ما عدا ان علاقات الاسبقية بين زوج الرموز مثلا (Y.X) يمكن ايجادها بواسطة YB∝X ، حيث ∝ هي (m – 1) من الرموز الموجودة الى اليسار B, X هي (n – 1) من الرموز الموجودة الى يمين Y ، ⌈m هي سلسلة الرموز الداخلة في المقدمة ، n هي سلسلة الرموز الداخلة بعد M⌋ .

وتوجد طريقة اخرى ايضا تتخذ القرار حول صحة النص أو عدم صحته استنادا الى درجة اسبقية العلاقات بين سلاسل الرموز تسمى بطريقة الـ (Bounded Context parser) (4) .

### 3 . 2 . 2 طرق الـ TOP – BOWN

كما بينا سابقا بأن محلل هذا النوع من الطرق يبدأ عملية الاعراب من جذر التفرع (ROOT) ويعمل الى ان يجد الهدف GOAL اي يبدأ من يسار النص والى يمينه (اذا كان النص مكتوب من اليسار الى اليمين) .

منها طرق الـ (LL – Parsers ، وسوف نشرح احدى هذه الطرق وهي .

### 3 . 2 . 2 . 1 طريقة الانحدار التكراري الذاتي Recursive Descent (2)

في هذه الطريقة يستخدم روتين او سلسلة اجراءات (Procedure) لتمييز كل رمز غير نهائي Non Terminal – في التفرع TREE فعلى سبيل المثال لو كانت لدينا القواعد التالية : –

| | | | |
|---|---|---|---|
| A ⟶ | (B) | (C) | X |
| B ⟶ | AB | A | |
| C ⟶ | Ay | Y | |

فأن الرمز A يقودنا الى احدى السلاسل التالية : –

X

(y)

(Ay)

(AA.....A)

اذن يمكن وضع روتين او سلسلة اجراءات معينة لانجاز اعراب الرمز غير النهائي (A) . وكما سنرى في البند اللاحق بأن هذه الطريقة ملائمة جدا لطبيعة قواعد البرمجة العربية المستحدثة (ليث) .

**4 – اختبار طريقة الاعراب المناسبة للغة البرمجة العربية المستحدثة (ليث)**

من دراسة التركيب العام للغة البرمجة العربية المستحدثة (ليث) يتبين لنا بأن اللغة هي من نوع (1) وفقا للاستنتاج التالي : –

نعلم ان بعض القواعد التي يستخدم لها طرق الـ TOP – Down في عملية الاعراب تسمى بالـ(LL)

فاذا استخدم المحلل رمز واحد فقط من النص في العملية فأن القواعد بالـ (1) واذا استخدم رمزين فالقواعد تسمى بالـ (2) ، LL ، وبصورة عامة اذا استخدم K الرموز فالقواعد تسمى بالـ LL (K) والرمز الذي يستخدمه المحلل لاجراء عملية الاعراب بالقواعد الـ (1) LL يجب ان تتوفر فيه الشروط التالية : –

أ – ان يكون رمزا غير نهائي Non – Terminal

ب – ان يكون في اقصى اليسار من النص (الجملة) .

حيث ان اتخاذ القرار حول صحت النص يكون استنادا الى الاحتمالات الممكنة التي سوف يستبدل بها هذا الرمز ، اي ان كل رمز غير نهائي يجب ان يقود الى تعبير (Expansion) واحد فقط

لرمز غير نهائي ، وعلى هذا الاساس فأن لكل تعبير معين توجد مجموعة واحدة فقط من الرموز النهائي والتي تسمى بالرموز الموجهة (Director – Symbols) فعلى سبيل المثال لو كان لدينا التعبير التالي للرمز (A)

حيث

$$A \longrightarrow \alpha_1 | \alpha_2 | \alpha_3 \quad | \ldots\ldots | \alpha_n$$

A هو رمز غير نهائي .

n سلسلة من الرموز (String)

ان الرموز الموجهة الـ $\alpha_i$ تحتوي على كل الرموز النهائية التي ممكن ايجادها في يسار اية سلسلة تولد بواسطة $\alpha_i$، هذه المجموعة من الرموز الشروع او الانطلاق (Starter – Symbols)$S(\alpha_i)$ والتي تعرف بالصيغة التالية : –

حيث : –

$$S(\alpha) = a \in Ve$$

$$S(\alpha) = a \in Ve \mid \alpha \Rightarrow^{*} aB,\ B \in (VtuVn)^{*}$$

S رمز الانطلاق Starter – Symbol

Vt مجموعة الرموز النهائية Terminals

Vn مجموعة الرموز غير النهائية Non – Terminals

B سلسلة من الرموز String

$\alpha$ سلسلة من الرموز $(\alpha \in (VtUVn)^{*})$

علما بأن ليس من الضروري ان تكون رموز الشروع كل مجموعة الرموز الموجهة وذلك لان $\alpha_i$ ممكن ان تولد سلسلة فارغة من الرموز (Empty – String) والتي هي ليست رموز الشروع ، لنرى ماذا يحدث في مثل هذه الحالة بعد سلسلة من التعابير (Expansion) والتي تبدا بالقاعدة Z التالية : –

$$Z \longrightarrow \quad - - - - \; BAS$$

حيث : –

A رمز في مقدمة النص (الهدف)

B سلسلة من الرموز التي من ضمن $Vt^{*}$ ($Vt^{*}$ وهي مجموعة من الرموز النهائية ومن ضمنها الرموز الفارغة) .

S سلسلة الرموز التي من ضمن $(VtUVn)^{*}$

فأذا كان التعبير A يقودنا الى سلسلة فارغة من الرموز فأن الرمز الموجود على يسار نص المصدر (Source – Text) هو رمز الشروع لـ S ، ولكن بما ان رموز الشروع S .

هي من ضمن الرموز الموجهة للتعبير والذي يقودنا الى سلسلة فارغة من الرموز ، ولكن يوجد واحد فقط مثل هذا التعبير ، لذلك فأن رموز الشروع S سوف توجد في اكثر من مجموعة من الرموز الموجهة ، وفي هذه الحالة فأن المحلل ليس بأستطاعته اتخاذ القرار لاي تعبير يطبق .

رموز الشروع لكل السلاسل والتي ممكن ان تتبع A تسمى بالتوابع (FOLLOWERS) والتي تعرف بالصيغة التالية :

$$F(A) = \{aZ \text{—} BAS,\ Z \text{ the axion},\ B,\ S \not\in (VtUVn)^*,\ a \in S(S)\}$$

حيث : –

A اي رمز غير نهائي Non – terminal

والان ممكن ان تعرف الرموز الموجهة لاي تعبير $\alpha$ للرمز غير النهائي A بالصورة التالية : –

$$DS(A,\alpha) = \{ a/a\ S(\alpha) \underline{\text{or}} (\alpha \xrightarrow{*} \in \underline{\text{and}}\ a \in F(A)) \}$$

خلاصة لذلك كل رمز غير نهائي ممكن التعامل معه (الوصول اليه) من القاعدة (Axiom) (والذي يمكن ايجاده الاقل في سلسلة واحدة تولد من القاعدة) ، وكل تعبير ممكن ان يقودنا على الاقل الى سلسلة واحدة لا تحتوي على رموز .

وبذلك استطعنا ان نعطي الشرط اللازم والكافي لقواعد LL (1) والذي هو بأن الرموز الموجهة والعائدة للتعابير المختلفة للرموز غير النهائية يجب ان تكون مجموعات منفصلة ، اما التبرير لهذا الشرط فيكون بالصورة التالية : –

أ – الشرط ضروري وذلك لانه ممكن ايجاد الرمز في مجموعتين من الرموز الموجهة وبذلك فأن المحلل لا يستطيع ان يتخذ القرار حول اي تعبير يطبق وبدون معلومات اضافية .

ب – الشرط كافي وذلك لان المحلل يستطيع ان يختار التعبير بدلالة الرمز المعطى وهذا الاختيار دائما هو الصحيح ، حيث انه اذا كان الرمز غير موجود من ضمن الرموزالموجهة فأن نص المصدر غير موجود في اللغة .

وبذلك لو قارنا بين التركيب العام للغة البرمجة العربية العليا (ليث) والشروط السابقة اعلاه لرأينا ان اللغة تخضع لقواعد الـ LL (1) ، حيث ان كل تراكيب الجمل تبدأ بكلمة محجوزة ، (Reserved Word) وهذه الكلمة تقودنا الى تركيب اخر يحل محل هذه الكلمة المحجوزة مثل : –

< ايعاز > : = < اقرأ > | < اطبع > | < افتح > | < اكتب > | < اضف > ...

< أقرأ > : = < اسم ملف > · < اسم ملف > عند الانتهاء < ايعاز > .

< اطبع > : = < اسم مفيد > .