الفصل السادس

الإعراب من الأسفل إلى الأعلى

BOTTOM-UP PARSING

- ما هو الإعراب من الأسفل إلى الأعلى
- المعرّب Shift Reduce Parser
- المعرّب من اليسار إلى اليمين LR Parser
- تمارين

الفصل السادس: الإعراب من الأسفل إلى الأعلى

6.1 ما هو الإعراب من الأسفل إلى الأعلى (Bottom up parsing)

يعرف الإعراب من الأسفل إلى الأعلى على انه محاولة لتقليص الجملة المدخلة w للوصول إلى رمز بداية القواعد باستخدام الاشتقاق أقصى اليمين للجملة w بالعكس (Right Most Derivation in Reverse) وهذا مكافئ لإنشاء شجرة إعراب للجملة المدخلة w وذلك بالبدء من أوراق الشجرة القواعدية (leaves) والتقدم باتجاه قمة الشجرة (root) أي محاولة إنشاء شجرة إعراب من الأسفل إلى الأعلى وهذا يتضمن البحث عن جزء من سلسلة الإدخال (substring) الذي يطابق جهة اليمين من كل قاعدة إنتاج للقواعد المعطاة ثم يستبدل ذلك الجزء من السلسلة بما موجود في جهة اليسار من قاعدة الإنتاج والتي هي عبارة عن رمز قابل للاشتقاق وهذه العملية تسمى بالتقليص (Reduction) التي هي ليست أكثر من عملية اشتقاق بالعكس. إن السبب الذي يجعل طريقة الإعراب من الأسفل إلى الأعلى أن تستخدم الاشتقاق أقصى اليمين للجملة w بالعكس وليس الاشتقاق أقصى اليسار هو لأن المعرّب يقرأ الجملة المراد إعرابها من اليسار إلى اليمين على شكل مقطع في كل مرة ويتتبع الاشتقاق أقصى اليمين بالعكس كما سنرى في الأمثلة اللاحقة.

6.2 المعرّب Shift Reduce Parser

إن هذا المعرّب يقوم بالإعراب من الأسفل إلى الأعلى أي يبدأ من الجملة وينتهي برمز البداية ومستخدما في عمله الاشتقاق أقصى اليمين بالعكس(Right Most Derivation in Reverse) وفي هذا المعرّب يتم استخدام المكدس ليتم وضع الرموز فيه إلى أن يحصل تطابق بين ما موجود في اعلي المكدس مع المدخلات المراد إعرابها فيتم إخراج ما موجود في المكدس والانتقال إلى المدخل الثاني من المدخلات كما سيتم الشرح في المثال التالي :

مثال6.1: لو كانت لدينا القواعد الآتية:

$E \rightarrow E+E \mid E*E \mid E-E \mid E/E \mid (E) \mid id$

وكانت لدينا الجملة التالية :

W= **id + id*id$**

فان المعرّب يقوم بإعراب الجملة فيما إذا كانت صحيحة قواعديا أم لا وكما يلي :

إن هذه الطريقة تحتاج في عملها إلى مكدس كهياكل بيانات ومدخلات وتحركات تشمل عمليتي الـ Shift والـ Reduce وفي البداية يوضع الرمز $ في المكدس وفي المدخلات توضع الجملة المراد إعرابها ، كما تحتاج هذه الطريقة إلى الاشتقاق أقصى اليمين للحصول على مقاود التقليص وكما يلي :

$E \rightarrow \underline{\mathbf{E+E}} \rightarrow E+\underline{\mathbf{E*E}} \rightarrow E+E*\underline{\mathbf{id}} \rightarrow E+\underline{\mathbf{id}}*id \rightarrow \underline{\mathbf{id}}+id*id$

وهكذا تم الحصول على مقاود التقليص التي تحتها خط في اشتقاق أقصى اليمين أعلاه. إن خوارزمية هذا المعرّب تنص على انه يتم الاستمرار بوضع المدخلات في المكدس لحين الحصول على مقود تقليص ليتم إرجاعه إلى أصله حتى نصل في النهاية إلى رمز البداية وتكون الجملة صحيحة قواعديا وإلا فلا تكون مقبولة قواعديا. ففي الخطوة الأولى من عمل الخوارزمية في هذا المثال سيتم وضع المدخل في المكدس بعملية الـShift id ، إما في الخطوة التالية فنجد أن هناك مقود تقليص أصبح في المكدس يرجع الى أصله بعملية الـ Reduce واستخدام قاعدة الإنتاج $E \rightarrow id$ لذا يستبدل الرمز id الذي كان موجود في المكدس بـ E وهكذا تستمر العملية إلى إن تنتهي المدخلات ولا يبقى شيء في المدخلات عدا رمز الـ $ الذي يشير إلى نهاية المدخلات لنجد رمز البداية في المكدس E لذا فان الجملة المطلوبة **id + id*id$** مقبولة قواعديا في اللغة المقابلة للقواعد أعلاه .

المكدس	المدخلات	التحركات
$	id + id*id$	Shift id
$id	+ id*id$	Reduce by E→ id
$E	+ id*id$	Shift +
$E+	id*id$	Shift id
$E+id	*id$	Reduce by E→ id
$E+E	*id$	Reduce by E→ E+E
$E	*id$	Shift *
$E*	id$	Shift id
$E*id	$	Reduce by E→ id
$ E*E	$	Reduce by E→ E*E
$E	$	مقبولة

مثال6.2: لو كانت لديك القواعد التالية:

S→AA

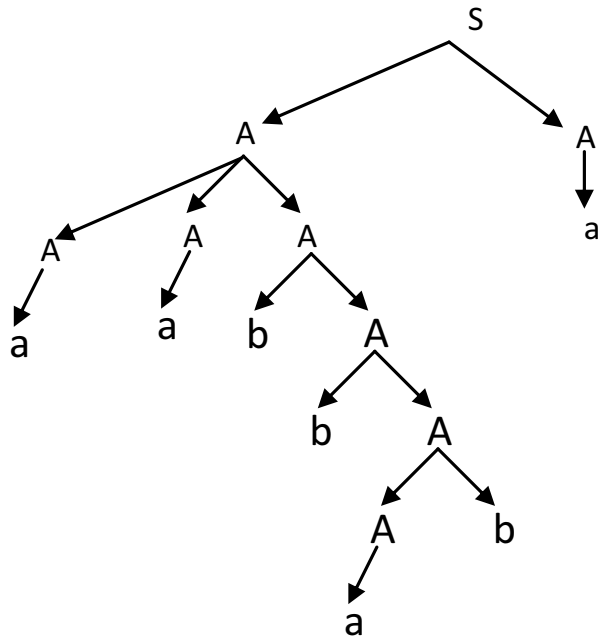
A→AAA | bA| Ab| a

هل أن الجملة **$aabbaba** مقبولة في اللغة المقابلة للقواعد أعلاه مستخدما شجرة الإعراب في الشكل (6.1) وطريقة إعراب الـ shift reduce ؟

إذن الجملة مقبولة في اللغة المقابلة للقواعد أعلاه حسب شجرة الإعراب السابقة .

الآن لو استخدمنا الاشتقاق أقصى اليمين للجملة المراد إعرابها لحصلنا على ما يلي:

E → **AA** →A**a** →**AAA**a →AA**bA**a →AAb**bA**a →AAbb**Ab**a →AAbb**a**ba →A**a**bbaba →**a**abbaba



شكل (6.1) شجرة الإعراب للجملة aabbaba

وبذلك نكون قد حصلنا على مقاود التقليص التي تحتها خط في الاشتقاق السابق والتي ستستخدم في خوارزمية الإعراب وكما يلي :

المكدس	المدخلات	التحركات
$	aabbaba$	Shift a
$a	abbaba$	Reduce by A→a
$A	abbaba$	Shift a
$Aa	bbaba$	Reduce by A→a
$AA	bbaba$	Shift b
$AAb	baba$	Shift b
$AAbb	aba$	Shift a
$AAbba	ba$	Reduce by A→a
$AAbbA	ba$	Shift b
$AAbbAb	a$	Reduce by A→Ab

$AAbbA	a$	Reduce by A→bA
$AAbA	a$	Reduce by A→bA
$AAA	a$	Reduce by A→AAA
$A	a$	Shift a
$Aa	$	Reduce by A→a
$AA	$	Reduce by S→AA
$S	$	الجملة مقبولة

مثال6.3: لو كان لدينا القواعد التالية:

E → E+T | T

T→ T*F | F

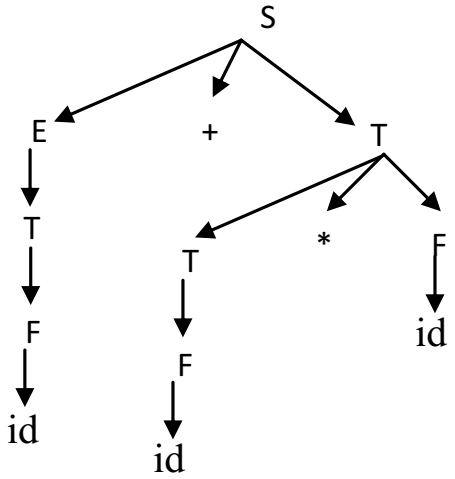
F→ id

وكانت لدينا الجملة التالية المراد إعرابها بطريقة (shift reduce)

W= id + id * id $

أولا يجب تنفيذ الاشتقاق أقصى اليمين وتوليد شجرة الإعراب الموضحة في الشكل (6.2) حيث أن طريقتي الإعراب أعلاه تحتاج إلى تنفيذ الاشتقاق أقصى اليمين ويتم استخدامه بالعكس (Right Most Derivation in reverse) كما سنرى لاحقا.

E → **E+T**→ E+**T*F**→ E+T***id**→ E+**F***id→ E+ **id***id→ **T**+ id*id →**F**+ id*id → **id** + id*id



شكل (6.2) شجرة الإعراب للجملة id + id * id

المكدس	المدخلات	التحركات
$	id + id*id$	Shift id
$id	+ id*id$	Reduce by F→ id
$F	+ id*id$	Reduce by T→ F
$T	+ id*id$	Reduce by E→ T
$E	+id*id$	Shift +
$E+	id*id$	Shift id
$E+id	*id$	Reduce by F→ id
$E+F	*id$	Reduce by T→ F

المكدس	المدخلات	التحركات
$E+T	*id$	Shift *
$E+T*	id$	Shift id
$E+T*id	$	Reduce by F→ id
$E+T*F	$	Reduce by T→ T*F
$E+T	$	Reduce by E→E+T
$ E	$	الجملة مقبولة

6.3 المعرّب من اليسار الى اليمين LR Parser

LR parser هو مختصر إلى (Left to right Right most derivation) أي الإعراب من اليسار إلى اليمين باستخدام اشتقاق أقصى اليمين. يحتاج هذا المعرّب في عمله إلى أتمتة محددة (Deterministic Finite Automata DFA) ولبناء هذه الأتمتة نحتاج إلى نوع آخر من القواعد يسمى (**Augmented grammar**) وهو إذا كان لدينا القواعد التالية:

G={ V, T ,P ,S}

فان القواعد المسماة بالـ (Augmented grammar) هي:

G1={ V∪ {S1}, T ,P∪ {S1→S} ,S1 }

أي تم إضافة قاعدة الإنتاج S1→S واعتبار S1 هو رمز البداية .