تصميم لغة برمجة عربية للاغراض التعليمية على نمط (BASIC)

جين خليل اسطيفان
المركز القومي للحاسبات الالكترونية

اولا : المقدمة

كانت اللغة العربية لغة الثقافة والعلم في أوج عظمة الحضارة العربية الاسلامية ، ولا تزال اليوم وهي تسير بخطى حثيثة الى الامام لمواكبة العصر والتطور التكنولوجي ، معززة اصالتها وقدرتها ووجودها الحي والفاعل بين الأمم الناهضة المتطورة .

وحيث ان الحاسبات الالكترونية تعتبر انجازا متقدماً للمسيرة التكنولوجية والعلمية بفعل الجهد البشري بحيث لم تسبق زاوية من زوايا الحاجات الانسانية المتعددة الا ودخلتها الحاسبات الالكترونية من أوسع أبوابها .

وحيث أن المستفيد يتعامل مع الحاسبات الالكترونية عبر اللغة التي تفهمها الحاسبة الالكترونية ، فكانت الحاجة الملحة تستدعي ايجاد لغة عربية تتعامل مع هذه التقنية يسهل تداولها من قبل المستفيد العربي .

يتضمن هذا البحث تصميم لغة برمجة عربية عليا مقتبسة من لغة (BASIC) بحيث يصبح من الممكن كتابة برامج في اللغة العربية المصممة وتدقيق هذه البرامج بواسطة محلل القواعد (SYNTAX ANALYZER) واعطاء جمل الاخطاء في حالة وجود خطأ . كذلك يتضمن البحث ترجمة هذه البرامج المكتوبة باللغة العربية الى لغة (BASIC) وخزنها على ملف ثم استدعاء مترجم للغة (BASIC) تمهيدا لتنفيذها .

ثانيا : اللغة LANGUAGE

بصورة عامة ، اللغة ومن ضمنها لغات البرمجة هي مجموعة من سلاسل (STRINGS) خطيه من الرموز تتولد وفق طرق معينة لتعريف اللغة ومن الطرق الاساسية لتعيين الاساليب اللغوية هي طريقة المولد(GENERATIVE SCHEME) لتوليد الجمل المسموح بها في اللغة وطريقة المشخص (RECOGNIZER) الذي يقرر ما اذا كانت الجملة صحيحة لغويا أم لا ، وقد استخدمت هذه الطريقة في البحث .

ثالثا : القواعد (GRAMMARS)

تعتبر القواعد احدى الأدوات المنهجية لتعيين لغة غير محدودة بطريقة محددة في شكل مجموعة من السلاسل ، ويطلق عليها أحيانا تعبير (PSG-PHRASE STRUCTURE GRAMMARS أي طرق التعبير عن أشباه الجمل .

والقاعدة (ق) عبارة عن نظام رياضي رباعي (سـ ، س ، ر ، أ) حيث :

سـ : ـ عبارة عن مجموعة محدودة من الرموز تسمى الرموز غير النهائية (NON-TERMINAL SYMBOLS)

س : ـ عبارة عن مجموعة محدودة من الرموز النهائية (TERMINAL SYMBOLS)

ر : ـ عبارة عن مجموعة محدودة من ازواج القواعد (RULES) بحيث ان كل قاعدة عبارة زوج مرتب من (أ ، ب) حيث « أ » هي سلسلة من الرموز النهائية وغير النهائية في (سـ ∪ س)$^{+}$ ويرمز لها (ف$^{+}$)^{0} في الجانب الايمن و « ب » عبارة عن

سلسلة من الرموز النهائية وغير النهائية في (ســ ∪ س)* ويرمز لها (ف*)° في الجانب الأيسر .

أي إن

أ ∈ (ســ ∪ س +) و ب ∈ (ســ ∪ س)*

وتكتب القاعدة بهذا الشكل أ ← ب

أ : ـ عبارة عن رمز مميز يسمى رمز الانطلاق (STARTER SYMBOL) ويعتبر أحد عناصر مجموعة الرموز غير النهائية «ســ» .

ومن هنا تعرف اللغة باستخدام أساليب التعبير عن أشباه الجمل (PSG) على أنها عبارة عن مجموعة سلاسل الرموز النهائية (TERMINAL SYMBOLS) التي تكونت كنتاج لتطبيق هذه القواعد .

1-3 أصناف القواعد TYPES OF GRAMMARS

جومسكي (CHOMSKY) هو عالم من أصل هولندي كان له فضل كبير في وضع قواعد ومنهجية رياضية يتم على أساسها تحليل اللغات وتعريفها باسلوب علمي وقد صنف اللغات الى أربعة انواع .

1 ـ النوع غير المحدود (UNRESTRICTED TYPE)

وأطلق عليه جومسكي صنف - ٥ (CHOMSKY TYPE - O)

°ف+ : المجموعة التي تحتوي على كل مفردات اللغة عدا السلسلة الفارغة (3)

ف+ : المجموعة التي تحتوي على كل مفردات اللغة وبضمنها السلسلة الفارغة (3)

ويشمل المجموعة المحدودة لأزواج قواعد التعبير عن أشباه الجمل (PSG) مثال من اللغة العربية المعرفة : ـ

< جملة ارجع > : : = « ارجع »

ويتميز هذا النوع من القواعد (جومسكي صنف ـ ٠) بامكانية اعطاء سلسلة فارغة (EMPTY STRING) في الجانب الأيسر لأحد أزواج القواعد (RULES) .

2 ـ النوع المتأثر بسياق المعنى (CONTEXT - SENSITIVE TYPE)

وأطلق عليه جومسكي صنف ـ 1 (CHOMSKY TYPE-1) ويتضمن مجموعة أزواج قواعد التعبير عن أشباه الجمل (PSG) التي على شكل ب ← جـ أي ان : ـ

$$\overline{ب}\ \overline{أ}_1 \leftarrow \overline{ب}\ \overline{أ}_2$$

$$\overline{جـ}\ \overline{أ}_1 \leftarrow \overline{ع}\ \overline{أ}_2$$

حيث بـ احد عناصر مجموعة الرموز غير النهائية (NON- TERMINAL SYMBOLS) $\overline{أ}_1$ و $\overline{أ}_2$ سلسلة من الرموز النهائية وغير النهائية في ($ف^*$)

عَ : سلسلة من الرموز النهائية وغير النهائية في ($ف^+$)

مثال من اللغة العربية المعرفة :

< جملة اذهب روتين > : : = « اذهب روتين » <LABEL>

< جملة ارجع > : : = « ارجع »

من هذا المثال يتبين ان جملة « اذهب روتين » يجب أن تظهر معها جملة « ارجع » وعدم ظهورها يعني ذلك ان القواعد ليست من صنف -1 .

ويمكن اعتبار أي قاعدة من نوع جومسكي صنف ـ 1 من ضمن قواعد جومسكي صنف ـ 0

3 ـ نوع حر السياق اي لا يتأثر بالمعنى (CONTEXT- FREE TYPE)

واطلق عليه جومسكي صنف ـ 2 (CHOMSKY TYPE- 2) وهو النوع الغالب لقواعد لغات الحاسب الالكتروني .

ويحتوي على مجموعة ازواج قواعد أشباه الجمل (PSG) التي من نوع جـ ← أ حيث : ـ

أ) جـ عنصر واحد وهو احد عناصر المجموعة غير النهائية .

ب) أ سلسلة من الرموز النهائية وغير النهائية .

حيث : ـ

$\overline{أ} \ni ف^*$

هذا النوع مغاير لنوع جومسكي صنف -1 حيث يمكن للرمز غير النهائي أن يكون سلسلة فارغة (EMPTY STRING) كما ان الطرف الايمن من زوج القاعدة (RULE) يحتوي على رمز غير نهائي (NON-TERMINAL SYMBOLE) واحد .

وقد صنفت اللغة العربية المصممة ضمن جومسكي صنف -2 .

مثال من اللغة العربية المعرفة : ـ

< جملة نسق > : : = « نسق » < DIMSPEC LIST >

ويمكن اعتبار أي قاعدة من نوع جومسكي صنف -2 من ضمن قواعد جومسكي صنف -1 .

4 ـ النوع المنتظم (REGULAR TYPE)

واطلق عليه جومسكي صنف -3 (CHOMSKY TYPE -3)

ويمكن تقسيمه الى صنفين :

أ ـ صنف خطي ايمن (RIGHT LINEAR TYPE)

وتكون مجموعة قواعد أزواج أشباه الجمل (PSG) لهذا الصنف من نوع

بـ ← ص جـ او بـ ← ص

حيث بـ ، جـ عناصر من المجموعة غير النهائية (NON-TERMINAL SYMBOLS) وص سلسلة من الرموز النهائية (ص ∈ س*)

مثال من اللغة العربية : ـ

< اذهب الى > : : = « اذهب الى » < LABEL >

< جملة نهاية > : : = « نهاية »

ب ـ صنف خطي ايسر (LEFT LINEAR TYPE)

وتكون مجموعة قواعد ازواج أشباه للجمل (PSG) لهذا الصنف من نوع

بـ ← جـ ص او بـ ← ص

ويدخل تصنيف قواعد هذا النوع (أي النوع المنتظم) ضمن قواعد النوع الحر السياق اي جومسكي صنف -2 .

رابعا : الاعراب PARSING

الإعراب أو تحليل التركيب (SYNTAX ANALYSIS) هو عملية تحديد اي قاعدة يمكن استخدامها في تكوين جمل اللغة او بتعبير اخر مهمة الاعراب هو اختبار صحة الجمل لمعرفة مدى تطابق تركيبها اللغوي مع قواعد اللغة المصممة لها .

1-4 طرق الاعراب

بصورة عامة يمكن تقسيم طرق الاعراب الى ثلاثة اقسام رئيسية فالقسم الاول يسمى بالطرق غير الحتمية (NON-DETERMINISTIC) والقسم الثاني يسمى بالطرق الحتمية (DETERMINISTIC) والقسم الثالث يسمى طرق التعقب الرجعي المحدد (LIMITED BACKTRACKING) .

1-1-4 الطرق غير الحتمية NON DETERMINISTIC

في هذه الطرق يقوم المحلل باختبار عدة اختيارات من القواعد (RULES) ثم الرجوع الى النقطة التي تم الاختيار الأول عندها في حالة عدم الوصول الى مطابقة النص الداخل (INPUT TEXT) .

2-1-4 الطرق الحتمية DETERMINISTIC

في هذه الطرق يقوم المحلل باختبار اختيار واحد والذي يقوده وبصورة مباشرة لمعرفة مطابقة النص الداخل (INPUT TEXT) مع القواعد المصممة ام لا دون

الرجوع الى نقطة الاختيار الأولى (NO-BACKTRACKING) .

هذه الطرق تعتبر أكثر كفاءة من الطرق غير الحتمية ولكنها تحتاج الى قواعد معينة غير مبهمة (UNAMBIGUOUS) .

مثل

1 ـ قواعد ـ LL (K)

2 ـ قواعد ـ LR (K)

3 ـ القواعد المسبقة PRECEDENCE GRAMMARS

وسنتطرق الى شرح طبيعة قواعد LL (K) نظرا لاستخدامها في اعراب البرامج المكتوبة باللغة العربية المصممة .

1-2-1-4 اعراب ـ LL (K)

يعتمد هذا النوع من الاعراب على طريقة التحليل من الأعلى الى الاسفل (TOP-DOWN) مع التنؤ المسبق في اختيار القواعد (RULES) وعلى أساس (ك) رمز من الجملة الداخلة حيث يوضع مؤشر يؤشر الى بداية اول رمز مع مشاهدة (SCAN) الرموز الجانبية على اليسار لاكتشاف الرموز التي تقع بجانب الرمز الأول .

3-1-4 ـ التعقيب الرجعي المحدد LIMITED BACKTRACKING

هذا النوع من التقنية يتضمن الرجوع ولكن كمية الرجوع تكون محدودة ونتيجة لذلك فالاعراب لهذه التقنية يكون أكثر اقتصاديا وأكثر كفاءة من طرق الاعراب غير الحتمية .

وتصنف طرق كل قسم الى نوعين : ـ

1 ـ الاعراب من الأعلى الى الاسفل (TOP-DOWN PARSING)

هو عبارة عن إيجاد سلسلة من القواعد تبدأ بالعنصر المميز أو رمز الانطلاق (STARTER SYMBOL) والمحاولة بالاستمرار في ايجاد القواعد التي تعطي الجملة الداخلة . او بعبارة اخرى اعراب هيكل الشجرة (TREE STRUCTURE) والبدء من الجذر (ROOT) والانتهاء بالاوراق (LEAF) .

2 ـ الاعراب من الاسفل الى الأعلى BOTTOM-UP PARSING

هو عبارة عن ايجاد سلسلة من القواعد تبدأ من الجملة الداخلة (عناصر مجموعة الرموز النهائية) وتستمر في الاختزال الى أن تصل العنصر المميز أو رمز الانطلاق (STARTER SYMBOL) او بعبارة اخرى أعراب هيكل الشجرة والبدء من الأوراق (LEAF) والانتهاء بالجذر (ROOT) .

خامسا : (BNF) BACKUS NORMAL FORM

استخدمت لغة BNF كلغة علمية لوصف قواعد لغات البرمجة وقد استخدمت هذه اللغة لاول مرة في وصف لغة ALGOL 60 والتي تعتبر الاساس في وصف أكثرية لغات البرمجة .

تستخدم هذه اللغة الرموز الوسيطة التالية :

1) : : = التي تعني احلال

مثلا

< جملة ارجع > : : = « ارجع »

أي <جملة ارجع >التي هي رمز غير نهائي (NON-TERMINAL SYMBOL)

معرفة بكلمة « ارجع » التي هي رمز نهائي (TERMINAL SYMBOL)

2) | التي تعني او

مثلا : -

<SIGN> ::= + | -

تعريف الاشارة اما بعلامة موجب (+) او بعلامة سالب (-) .

3) < > التي تعني ان الرمز غير نهائي (NON-TERMINAL SYMBOL) وقد استخدمت في تعريف اللغة العربية موضوع البحث لغة (BNF) الموسعة (EXTENDED BNF) التي تستخدم بالاضافة الى الرموز الثلاث المذكورة الرموز التالية : -

4 - (« . . . ») تعني تكرار الى التعبير السابق .

مثلا

<NUM LEFT PART> ::= <NUMERIC VARIABLE> = (. . . « + <NUMERIC VARIABLE> »)

د ع س + ص + ك

5 - { } التي تعني انه يمكن استخدام أي من التعابير الواردة داخل الأقواس .

مثلا

[<SIGN>] = : : < CONSTANT >

{ < FIYED > | < INTEGER > }

أي إن الرقم أما أن يكون عدد صحيح (INTEGER) او عدد فيه فاصلة عشرية (FIXED) .

6 - [ ] التي تعني احتواء الجملة على التعبير الوارد داخل القوس المربع اختياري .

مثلا : -

[< DIGIT >]< LETTER = : : < SIMPLE VARIARE >

أي ان المتغير البسيط عبارة عن حرف قد يكون متبوع برقم او لا .

سادسا : - اللغة العربية المصممة

اللغة العربية المعرفة شبيهة بلغة (BASIC) وقد تم حتى الان تعريف المصطلحات الاساسية للغة وهي (خمسة عشر مصطلح) وفي المستقبل يمكن استكمال تعريف باقي المصطلحات الفرعية .

1-6 شرح تفصيلي عن المصطلحات

(ملحق رقم 1 -)

1 - جملة ملاحظة REMARK STATEMENT

هذه الجملة تسمح للمبرمج أن يدون بعض الملاحظات لغرض توثيق فقرات البرنامج . كما يمكن استخدامها كفاصل بين اجزاء البرنامج .

أمثلة على ذلك

10 « ملاحظة » هذا البرنامج يجري عملية الترتيب تصاعديا .

20 « ملاحظة »

2 ـ جملة دع LET STATEMENT

هذه الجملة تعني احلال المقدار المتغير او الثابت في جهة اليسار محل المتغير الواقع في جهة اليمين .

امثلة على ذلك

10 « دع » س = 3

أي إن العدد 3 يخزن في موقع مشار اليه بالمتغير س .

20 «دع » ك $ = « البداية »

أي إن كلمة البداية تحل مكان المتغير ك $

3 ـ قراءة البيانات READING DATA

تستخدم لقراءة البيانات الخاصة بالمتغيرات من نوع (المتغير البسيط) (SIMPLE VARIABLE) او المتغير الهامشي . (SUBSCRIPTED VARIABLE) او المتغير السلسلي (STRING VARIABLE) وفي هذه الحالة تعرف البيانات باستخدام جملة معطيات (DATA STATEMENT) .

4 ـ جملة معطيات DATA STATEMENT

تستخدم لتعريف المعطيات أو البيانات المتعلقة بالمتغيرات الواردة بالبرنامج والمعرفة بواسطة جملة اقرأ .

مثال (1)

10 « اقرأ » س ، ص

20 « معطيات » 15.36

مثال (2)

10 اقرأ ل ، ب

20 « معطيات » « برنامج » ، « اللغة »

5 ـ جملة ادخل INPUT STATEMENT

تستخدم جملة ادخل لادخال معطيات المتغيرات من خلال الأنظمة الآنية المباشرة (ON-LINE TIME SHARING SYSTEM) .

6 ـ طبع البيانات PRINTING DATA

جملة اطبع تستخدم لطبع قيم المتغيرات او لطبع سلسلة حروف .

أمثلة :ـ

10 « اطبع » س* (ص + 2) ، ب 2 ، ك

20 « اطبع »

30 « اطبع » ل ا ، « برنامج » ، د

7 ـ جمل السيطرة CONTROL STATEMENTS

إن جمل البرنامج تنفذ بالتسلسل حسب تسلسل كتابة البرنامج الا اذا جاء إيعاز ما وغير التسلسل ومن هذه الايعازات ما يلي : ـ

أ ـ اذهب الى GO TO STATEMENT

تستخدم لغرض التفرع غير الشرطي (UNCONDITIONAL BRANCH) الى احد الايعازات التنفيذية الواردة قبلها أو بعدها ولا يسمح بالتفرع الى أحد الايعازات غير المنفذة (NON-EXECUTABLE STATEMENTS) مثل : ـ

إيعاز « ملاحظة » ، إيعاز « نسق » ، إيعاز « معطيات »

ب ـ جملة اذا ـ انفذ IF-THEN STATEMENT

تستخدم لغرض التفرع الشرطي (CONDITIONAL BRANCH)

اعتمادا على نتيجة الاحتساب والمقارنة .

مثال : ـ

10 « اذا » س = 10 « انفذ »200

8 ـ التكرار LOOPING

التكرارية هي من الأساليب المهمة في البرمجة والتي تعمل على انجاز احد الايعازات او مجموعة من الايعازات عدة مرات بالتعويض بمعطيات جديدة في كل مرة . وقد استخدم مصطلحا « لكل والتالي » لغرض التكرارية في لغتنا ، حيث انه التكرار (LOOP) يبدأ بجملة لكل وينتهي بجملة التالي .

شروط التكرار

1 ـ كل جملة لكل (FOR) تكون مرتبطة بجملة التالي (NEXT) الخاصة بها وتستخدم لتحديد بداية ونهاية التكرار .

2 ـ عدد الجمل بين لكل والتالي غير محددة .

3 ـ النقل من داخل التكرار (LOOP) الى خارج التكرار مسموح به بواطة جملة (اذا ـ اذا) وجملة (اذهب الى) ولا يسمح بالتفرع من الخارج الى داخل التكرار (LOOP) .

4 ـ جملة التكرار ممكن أن تكون شبكية (NESTED) اي ممكن أن تحتوي على جملة تكرار أخرى .

5 ـ في حالة عدم استخدام جملة (بزيادة) في تعريف جملة (لكل) يجب أن تكون القيمة الاصلية للمعطيات اكبر من القيمة النهائية .

9 ـ المصفوفات والمتغيرات الهامشية ARRAY AND SUBSCRIPTED VARIABLES يمكن خزن المعطيات في شكل مصفوفات وتكون من نوعين : ـ

1 ـ مصفوفة من نسق ـ 1 ONE -DIMENSIONAL ARRAY وتعطى قائمة LIST

2 ـ مصفوفة من نسق ـ 2 TWO -DIMENSIONAL ARRAY وتعطى جدول TABLE

10 ـ مصفوفة MATRIX

أ ـ جملة اقرأ مصفوفة

تستخدم لقراءة معطيات عناصر المصفوفة .

ب ـ جملة اطبع مصفوفة

تستخدم لطبع عناصر المصفوفة .

11 ـ البرامج الفرعية GOSUB

البرامج الفرعية تستخدم للتعبير عن مجموعة الايعازات التي قد يتكرر استخدامها عدة مرات في البرنامج الرئيسي وفي هذه الحالة يمكن استدعاؤ ها عند الحاجة اليها بواسطة مصطلح « اذهب روتين » وكل برنامج فرعي يجب أن ينتهي بمصطلح « ارجع » لتفريع البرنامج الى الجملة التي تلي جملة الاستدعاء .

12 ـ النهاية END

كل برنامج يجب أن ينتهي بجملة « نهاية » END

وتبدأ الجملة وفقا لتعريف هذه اللغة برقم محصور بين العدد 1 - 99999 بحيث لا يزيد عن خمسة مواضع رقمية ويكون متبوعا بمصطلح من مصطلحات اللغة ثم يلي ذلك باقي رموز الجملة كما سبق تعريفها .

ويفصل بين عناصر الجملة فراغ واحد على الاقل (DELIMITER) ويسمح باحتواء احد رموز الجملة على الحرف فراغ (SPACE) عدا الرموز التي خصصت لتعريف المصطلحات نفسها فمثلا لا يجوز كتابة نسق وانما تكتب حروف الكلمة بدون ترك فراغات بينها (نسق) .

سابعا : البرنامج المترجم ABTRP

ARABIC BASIC TRANSLATION PROGRAM - : ABTRP

استخدمت لغة البرمجة العليا فورتران (FORTRAN) لكتابة البرنامج المترجم (ABTRP) وتمت تمشيته على الحاسبة CHB 66-10 ومهمة البرنامج المترجم تدقيق قواعد اللغة الخاصة بجمل البرامج التي تكتب بواسطة اللغة العربية المصممة .

1-7 ـ الأجزاء الرئيسية التي يتكون منها البرنامج المترجم ABTRP

1 ـ يقوم بقراءة جملة من ملف الادخال INARB

2 ـ تدقيق الجملة من ناحية القواعد (SYNTAX)

3 ـ يقوم بكتابة الجملة بعد تدقيقها على ملف (OTARB) واذا كانت الجملة خالية من الأخطاء القواعدية فتكتب أيضا على ملف اخر هو (OTBSC) تمهيدا لتنفيذها .

2-7 ـ عمل البرنامج المترجم ABTRP

يتكون البرنامج المترجم من : .

1 ـ برنامج رئيسي MAIN PROGRAM

2 ـ مجموعة برامج فرعية SUBROUTINES

1-2-7 ـ البرنامج الرئيسي MAIN PROGRAM

يبدأ البرنامج بقراءة ملف الادخال (INARB) ، المخزون فيه البرنامج المكتوب باللغة العربية ، حيث تكون القراءة سطرا بعد اخر وقد استخدمت

الحروف اللاتينية في كتابة البرامج لعدم امكانية استخدام الحروف العربية مباشرة وبواسطة ايعازات السيطرة (JOB CONTROL LANGUAGE) الخاصة بالحاسبة يتم تحويل الحروف اللاتينية الى الحروف العربية ثم يستدعى روتين (CONVER) لتحويل الجملة من جهة اليمين الى جهة اليسار لكي تسهل معالجة البرنامج بلغة فورتران FORTRAN .

بعد ذلك تدقق الجملة بحيث لا تزيد الكتابة عن عمود 72 واذا ظهر هناك أي رمز من عمود 73 الى عمود 80 فمعنى ذلك أن هناك خطأ في الجملة .

ثم تدقق رقم الجملة (LINE NUMBER) الذي يتراوح بين (1-99999) وبما أن الأرقام تكون باتجاه معاكس لذلك فلقد وضعت معادلة لقلب الارقام وكذلك استعملت ايعازات مثل (ENCODE و DECODE) التي تحول البيانات من (I-FORMAT) الى (A-FORMAT) وبالعكس . وتكون ارقام الجمل في تسلسل تصاعدي .

كذلك استعملت المتغيرات I و II كمؤشرات لتتبع حروف الجملة المراد تدقيقها ويبدأ المؤشر I بالقيمة واحد (INTIAL VALUE) ويزداد باضافة واحد الى القيمة السابقة مع قراءة كل حرف من حروف الجملة المدققة أما المؤشر II فيبدأ بالقيمة 1 ويزداد تصاعديا باضافة واحد الى القيمة السابقة مع كتابة كل حرف من حروف الجملة بعد ترجمتها في ملف (OIBSC) لتنفيذها بواسطة مترجم لغة (BASIC)

2-2-7 ـ البرامج الفرعية SUBROUTINES

أ ـ من الروتينات الفرعية الموجودة في مكتبة الحاسبة CHB 66-10 والمستخدمة داخل البرنامج :ـ

1- CONCAT :

يقوم بدمج سلسلتين من الرموز لانتاج سلسلة رموز واحدة .

2- KOMPCH :

يقوم بمقارنة سلسلتين من الرموز واعطاء الناتج في دالة ثم يجري اختبار الدالة في جملة شرطية (IF-STATEMENT)

ب ـ من البرامج الفرعية المكتوبة

1- LABEL

يقوم بتدقيق عنوان (LABEL) الايعاز المذكور بعد مصطلح « اذهب الى » و« اذهب روتين » و« اتذ » .

2- SMPVAR

يقوم بتدقيق المتغير البسيط الذي يكون مكونا من حرف او حرف متبوع برقم .

3- CONST

يقوم بتدقيق الثوابت التي يمكن أن تكون رقما صحيحا او كسريا من نوع FIXED مسبقا بعلامة (+) او (−) .

4- SUBSVR

يقوم بتدقيق المتغير الهامشي SUBSCRIPTED في حالة تعريف القائمة (LIST) او الجدول (TABLE) .

5- DIMSPC

يقوم بتدقيق نوع مصطلح « نسق » الذي إما أن يكون نسق -1 أو نسق -2 .

NUMEXP -6

يقوم بتدقيق التعابير الرياضية .

LITRST -7

يقوم بتدقيق سلسلة الرموز وعكسها بين علامتـي(" ") .

ثامنا : الاستنتاجات

يتضمن هذا الجزء توضيح مراحل المشروع وما تم انجازه وكذلك المعوقات التي جابهتنا في مرحلة التنفيذ وطرح بعض التعـديلات التـي يمـكن ادخالهـا على المشروع سواء للتغلب على المعوقات او لمعالجة بعض النقاط التي لم يتسـع وقـت البحث لأخذها في الاعتبار خلال المرحلة الحالية .

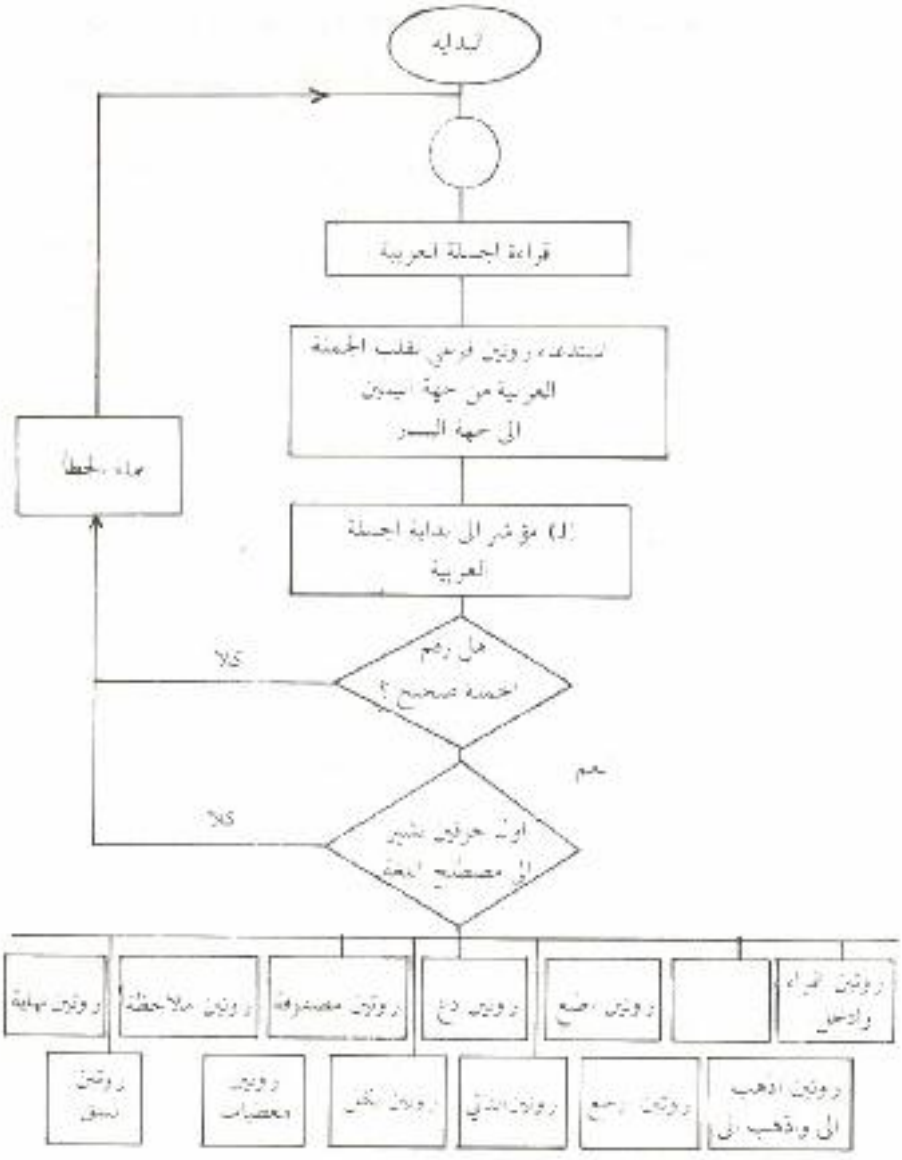
1 ـ المرحلة الأولى

شملت المرحلة الأولى من البحث تحديد قواعد لغة عربية علىا مقتبسة من لغة (BASIC, وكتابة برنامج مدقـق (SYNTAX, لهـا يلغـة (BASIC, الـذي استخـدم اسلوب المشخص (RECOGNIZER, في تدقيق اللغة وقد قام بتنفيذ هذه المرحلـة احد طلبة الدبلوم العالي ـ عبد الغني صالب وقد لاحظت ان البرنامج المدقق يحتوي على بعض النقص اذ لا يمكنه تدقيق كل القواعد المعرفة .

2 ـ المرحلة الثانية

وهي المرحلة التي قام كاتب البحث بتنفيذها ، وتشمل : ـ

اولا : اضافة بعض التعديلات على تعريف اللغة منها عدم تحديد عدد GOSUB ضمن برنامج محلل القواعد وكذلك تم اضافة قواعد أخرى منها : ـ



(المخطط الانسيابي العام لبرنامج المترجم)

(٥) يمكن مراجعة البحث الاصلي لتتبع المخطط الانسيابي التفصيلي للبرنامج المترجم

1 ـ في تعريف المتغير الرقمي (NUMERIC VARIABLE) تم اضافة البديل الثاني وهو المتغير الهامشي (SUBSCRIPTED VARIABLE) .

2 ـ اضافة القواعد الخاصة بمعالجة مصطلح « مصفوفة » .

3 ـ اضافة قواعد ثانية في تعريف التعبير (EXPRESSION) بحيث أصبح هناك نوعان من تعريف التعبير : ـ

أحدهما يتعلق بالتعبير الرياضي (EXPRESSION) الذي يدخل ضمن تعريف مصطلح « اذا » الذي يشمل العلاقات التالية : ـ

(< و > و = و <= و >=) والاخر يتعلق بالتعبير الرياضي (NUMERIC EXP.) الذي يدخل ضمن تعريف مصطلح « لكل » والذي لا يقبل العلاقات المذكورة سابقا .

ثانيا ـ كتابة البرنامج المترجم والذي يتضمن :

أ ـ مرحلة محلل القواعد (SYNTAX ANALYSER) لاكتشاف الاخطاء في البرامج المكتوبة بواسطة اللغة العربية المصممة . وقد تم تعديل بعض اجزاء البرنامج المدقق المعد في المرحلة الأولى لتلافي التكرار الوارد ومراعاة استخدام اسلوب تجزئة (MODULERIZATION) البرنامج لتسهيل متابعته وقد أعيد كتابة البرنامج مع التعديلات المقترحة بلغة فورتران (FORTRAN) نظرا لسيطرة الباحث عليها ولاعتبارات أخرى متعلقة بالحاسبة الالكترونية CHB 66-10

بالاضافة الى امكانية الاستفادة من البرامج الفرعية الموجودة في مكتبة (FORTRAN COMPILER) مثل KOMPCH. CONCAT .

ب ـ مرحلة الترجمة TRANSLATION

وتمر بها الجمل الخالية من الأخطاء اللغوية حيث تترجم عناصر الجملة من اللغة العربية الى لغة BASIC وتخزن على ملف OTBSC تمهيدا لتنفيذها بواسطة مترجم لغة BASIC .

المعوقات

أ ـ معوقات متعلقة بمكونات الحاسبة COMPUTER CONFIGURATIONS

وهي : الحروف اللاتينية : ـ

حيث استخدمت الحروف اللاتينية لكتابة برامج المستفيد (USER PROGRAMS) وكان من المفضل كتابتها مباشرة باللغة العربية لتسهيل التدقيق ولتحقيق الغرض الأساسي من تعريب لغة عربية للبرمجة وهو السماح للمستفيد بكتابة البرامج باللغة العربية . وقد أدى استخدام الحروف اللاتينية ثم تحويلها الى حروف عربية الى عدم امكانية استخدام بعض الحروف العربية مثل حرف أ ، بـ ، يـ ، لعدم وجود مقابل لها في الحروف اللاتينية .

ب ـ معوقات متعلقة بالمشروع نفسه : ـ

إن المشروع هو تكملة لمشروع احد طلبة الدبلوم العالي وان الوقت المخصص للبحث لم يكن كافيا لتعديل جميع العقبات المتعلقة بتنفيذ هذا المشروع ومن أهم هذه العقبات هي .

1 ـ عند تعريف اللغة BNF لم تتم ترجمة القواعد الخاصة بجميع الرموز غير النهائية (NON-TERMINAL SYMBOLS) مما يدعو المستفيد الى ضرورة معرفة اللغة الانكليزية لتفهم قواعد اللغة المصممة .

2 ـ هناك بعض الازدواجية (AMBIGUITY)* في تعريف بعض القواعد مثلا .

(SIMPLE VARIABLE) و (NUMERIC ARRAY ID) لها نفس التعريف (ملحق رقم ١) .

3 ـ في تعريف الرمز غير النهائي <LABEL> في BNF لغة BASIC الذي يظهر مع المصطلحات GOSUB و GO TO و IF الى تقبل الصفر ضمن رقم الجملة وهذا خطأ بالنسبة الى محلل قواعد لغة (BASIC) .

4 ـ في البرنامج المدقق (SYNTAX) وبعد تدقيق رقم السطر يبدأ تدقيق الجملة بتدقيق اول حرفين فقط من المصطلح مما يؤ دي الى الازدواجية في تحديد بعض القواعد المشار اليها .

مثال : ـ

المصطلحات « اذهب الى » و« اذا » كلاهما يبدأ بالحرفين « اذا » مما يؤ دي الى الازدواجية وكان من الأفضل تدقيق المصطلح باكمله لتقليل وقت التنفيذ .

وكان من الممكن التغلب على بعض هذه المعوقات لولا ضيق الوقت المخصص للبحث .

الاقتراحات

١ ـ معالجة بعض المصطلحات الاضافية التي لم تؤ خذ في الاعتبار في المرحلة الحالية مثل

CHAIN, STOP, CALL, CONVERT, RESTORE

وغير ذلك من المصطلحات الأخرى .

* AMBIGUITY : ـ مجموعة قواعد از راح انتباه الجمل PSG ممكن ان تعطي نفس اللغة باكثر من اشتقاق واحد باستخدام قواعد مختلفة (RULES) .

وكذلك اعادة كتابة قواعد بعض المصطلحات لتكون أكثر شمولية . أمثلة على ذلك جملة <دع> . : : = « دع » LET PART >[<LETM>]<

جملة اذهب روتين : : = «MULTI -BRANCH GOSUB SINGLE BRANCH GOSUB وغيرها من القواعد الأخرى الذي أخذ الجزء الأساسي منها لتسهيل عمل محلل القواعد لها . إضافة الى ذلك اقترح اضافة بعض الدوال الضرورية مثل SST و VAL و STR لمعالجة السلاسل وكذلك الدوال العلمية الضرورية مثل SQR و COS و SIN والى اخره من الدوال الأخرى .

2 ـ دراسة كتابة البرامج المتعلقة بمرحلة توليد الرموز (CODE GENERATION) والاستغناء عن مرحلة الترجمة (TRANSLATION) من اللغة العربية الى لغة BASIC قبل تنفيذ البرامج .

3 ـ ينتهي تدقيق الجملة في الموقع الذي يكتشف به أول خطأ ومن الممكن تعديل البرنامج بحيث يسمح بالاستمرار في عملية اكتشاف الاخطاء وحتى الانتهاء من تدقيق الجملة الداخلة بالكامل وذلك في الحالات التي تسمح بذلك .

4 ـ نقترح اجراء بعض التحويلات (TRANSFORMATION) على قواعد اللغة العربية تمهيدا لاستخدام اسلوب (FINITE AUTOMATA) في التدقيق نظرا لسهولته .

5 ـ دراسة استخدام اللغات الخاصة بكتابة المترجمات (COMPILER -COMPILER) كبديل للغات FORTRAN و BASIC لغرض تسهيل كتابة البرامج وسرعة التنفيذ مثل (BCL -BEST COMILER LANGUAGE) (CDL-COMPILER DESIGN LANGUAGE)

ملحق رقم ـ1 ـ

اللغة العربية المصممة BNF

(< FIXED > |<INTEGER>) (<SIGN>) ::< CONSTANT >

− \ + = :: < SIGN >

< FIXED > | < INTEGER > ::< UNSIGNED CONSTANT >

< DIGIT > ::< INTEGER >

9| 8 | 7| 6| 5| 4| 3| 2| 1| 0 ::< DIGIT >

أ | ب | ت | ث | ج | ح | خ | د | ذ | ر | ز | = :: < INTEGER >

س| ش | ص | ط | ظ | ض | ع | غ | ف | ق | ك | ل | م | ن | هـ |

و | ي | ى

< INTEGER > . | < INTEGER > . < INTEGER > = : : < FIXED >

< STRING VARIABLE > | < NUMERIC VARIABLE > = ::< VARIABLE >

< SUBSCRIPTED VARIABLE > | < SIMPLE VARIABLE > = : : < NUMERIC VARIABLE >

[< DIGIT > | < LETTER > = : : < SIMPLE VARIABLE >

| < SUBLIST > | < NUMERIC ARRAY ID > = : : < SUBSCRIPTED VARIABLE >

| < SUBSCRIPT >, | < SUBSCRIPT > = : : < SUBLIST >

< SIMPLE VARIABLE > = : : < SUBSCRIPT >

[< DIGIT >] < LETTER > = : : < NUMERIC ARRAY ID >

< STRING ARRAY VARIABLE > | < STRING SIMPLE VARIABLE > = : : < STRING VARIABLE >

$ [< DIGIT > LETTER > = : : < STRING SIMPLE VARIABLE >

(< SUBSCRIPT >) < STRING ARRAY ID > = : : < STRING ARRAY VARIABLE >

$ | < DIGIT >] < LETTER > = : : < STRING ARRAY ID >

.... < RELAT > = : : < NUMERIC EXPRESSION >

.... < MIN > = : : < RELAT >

.... [< SI >] < SI > = : : < MIN >

.... [< T > { - 1 + }] < T > < UNOP > = : : < SI >

.... [< FA > { ★ |] < FA > = : : < T >

.... [< PRIM > ★ ★] < PRIM > = : : < FA >

< NUMERIC EXPRESSION > | < UNSIGNED CONSTANT > | < NUMERIC VARIABLE > = : : < PRIM >

< RELATION > = : : < EXPRESSION >

| [< MINMAX > < RELATIONAL OPERATOR >] < MINMAX > = : : < RELATION >

< STRING EXPRESSION > < RELATIONAL OPERATOR > < STRING EXPRESSION >

... [< SUM >] | < SUM > = : : < MINMAX >

| < TERM > { - | + }] < TERM > < UNOP > = : : < SUM >

.... [< TERM > ★ (•)] < FACTOR > = : : < TERM >

.... [< PRIMARY > ★★] < PRIMARY > = : : < FACTOR >

- 1 + = : : < UNOP >

(EXPRESSION) | < UNISIGNED CONSTANT > | < NUMERIC VARIABLE > = : : < PRIMARY >

< > | = | > | < | = > | = < = : : < RELATIONAL OPERATOR >

< LET PART > ، دع ، = : : < جملة دع >

< STRING LET > | < NUM LET > = : : < LET PART >

< NUMERIC EXPRESSION > < NUM LEFT PART > = : : < NUM LET >

.... { = < NUMERIC VARIABLE > } = : : < NUM LEFT PART >

< STRING EXPRESSION > < STRING LEFT PART > = : : < STRING LET >

..... { = STRING VARIABLE > } = : : < STRING LEFT PART >

< SOURCE STRING > = : : < STRING EXPRESSION >

< LITERAL STRING > | < STRING VARIABLE > = : : < SOURCE STRING >

.... < CHARACTER > = : : < LITERAL STRING >

ANY ASCUI GRAPHI GHARACTER OTHER THAN» = : : < CHARACTER >

< CHARACTER STRING > « ملاحظة » : : = « ملاحظة » < جملة ملاحظة >

ANY ASCII GRAPHIC CHARACTER = : : < CHARACTER STRING >

< جملة اذهب الى > : : = « اذهب الى » < LABEL >

< INTEGER > = : : < LABEL >

< جملة اذهب روتين > : : = « اذهب روتين » < LABEL >

< جملة ارجع > : : = « ارجع »

< جملة نهاية > : : = « نهاية »

< جملة لكل > : : = « لكل » < FOR VAR > = < INTIAL VALUE > « الى » < FINAL BALUE > « بزيادة » [< ⊄⊃≠⌒∉ | $^{+}$≠ >]

< SIMPLE VARIABLE > = : : < FOR VAR >

< NUMERIC EXPRESSION > = : : < INTIAL VALUE >

< NUMERIC EXPRESSION > = : : < FINAL VALUE >

< NUMERIC EXPRESSION > = : : < STEP SIZE >

< FOR VAR > « التالي » = : : < جملة التالي >

< LABEL > < EXPRESSION > « اذا » = : : < جملة اذا >

[< PRINT LIST >] « اطبع » = : : < جملة اطبع >

.... [< PRINT ELEMENT >,]<PRINT ELEMENT >= : :<PRINT LIST >

| [< LITERAL STRING > | < NUMERIC EXPRESSION > = : : < PRINT ELEMENT > < VARIABLE >

< READ ITEM LIST > « اقرأ » = : : < جملة اقرأ >

... | < VARIABLE >,] < VARIABLE > = : : < READ ITEM LIST >

< DATA CONSTANT LIST > « معطيات » = : : < جملة معطيات >

< LITERAL STRING > | < CONSTANT > = : : < DATA CONSTANT LIST >

< READ ITEM LIST > « ادخل » : : = < جملة ادخل >

.... [DIMSPEC LIST,] < DIM SPEC LIST > « نسق » = : : < جملة نسق >

< STRING DIMSPEC > | < NUMERIC DIMSPEC > = : : < DIMSPEC LIST >

([< BOUND > ,] < BOUND >) < NUMERIC ARRAYID > = : : < NUMERIC DIMSPEC >

(< BOUND >) < STRING ARRAY ID > = : : < STRING DIMSPEC >

< O,TEGER > = : : < BOUND >

< MAT READ ITEM LIST > « مصفوفة اقرأ » = : : < جملة مصفوفة اقرأ >

... [< NUMERIC ARRAY ID > | < NUMERIC ARRAY ID > = : : < MAT READ ITEM LIST >

< MAT READ ITEM LIST > « مصفوفة ادخل » = : : < جملة مصفوفة ادخل >

< MAT PRINT ITEM LIST > « مصفوفة اطبع » = : : < جملة مصفوفة اطبع >

.... [< NUMERIC ARRAY ID >] < NUMERIC ARRAY ID > = : < MAT PRINT ITEM LIST >

< NUMERIC ARRAY ID > = < NUMERIC ARRAY ID >

« مصفوفة » = : : < جملة مصفوفة >

| < NUMERIC ARRAY ID > +

< NUMERIC ARRAY ID > = < NUMERIC ARRAY ID > «مصفوفة»

| < NUMERIC ARRAY ID > -

< NUMERIC ARRAY ID > = < NUMERIC ARRAY ID > «مصفوفة»

| < NUMERIC ARRAY ID > *

* < CONSTANT > = < NUMERIC EXPRESSION > «مصفوفة»

< NUMERIC ARRAY ID >

ملحق رقم – 2 –

مثال – 1 – INARB

```
10    ملاحظه - برنامج مفكوك
20          ابقا  ض
30          دع  ذ = 1
40          لكل ر= 1 الى ض
50             دع  ذ = ذ * ر
60          التالى ر
70          اطبع "مفكوك" ، ض ، ذ
80          معطيات ؟
90          نهايه
```

INARB - 2 - مثال

```
10          ملاحظه - مجموع (10) اعداد
20          حسق ت(10)
30          اقرا ن
40          لكل ر= 1 الى ن
50          اقرا ت(ر)
60          التالى ر
70          دع ف= 0
80          لكل س= 1 الى ن
90          دع ف=ف + ت(س)
100         التالى س
110         اطبع " مجموع "؛ف
120         معطيات 10
130         معطيات 1،2،3،4،5،6،7،8،9،10
140         نهايه
```

OTBSC

```
10          REM - WDWZQ )01( ;QG;G
20           DIM  X(10)
30           READ    N
40          FOR I= 1  TO  N
50             READ    X(I)
60          NEXT   I
70           LET   S= 0.
80         FOR K= 1  TO   N
90            LET   S=S + X(K)
100 NEXT   K
110       PRINT " QZWDW ",S
120          DATA   10
130          DATA 1,2,3,4,5,6,7,8,9,10
140           END
```

OTARB

```
10                ملاحظه - مجموع (10) اعداد
20                   دسق د(10)
10                   اقرا   ض
20                لكل ر= 1  الى ض
50                   اقرا   د(ر)
30                التالى  ر
40                دع   ف= 0
50             لكل س= 1  الى  ض
60                دع   ف=ف + د(س)
70             التالى  س
80                اطبع " مجموع "؛ف
90                معطيات  10
100            معطيات 1،2،3،4،5،6،7،8،9،10
110               نهايه
120            مجموع  الاخطاء  =
```

OTARB

198

```
ملاحظه - برنامج مفكوك
اقفا  ض
خطا فى اقرا او ادخل سطر---20
دع  خ = 1
لكل ر= 1 الى  ض
دع  خ = خ * ر
التالى ر
اطبع "مفكوك" ، ض ، خ
معطيات 5
نهايه
مجموع  الاخطاء  =          1
```

المصادر

أ ـ عبد الغني صائب عبد الغني ، لغة عربية عليا للبرمجة على نمط لغة BASIC كانون الاول 1976 .

2. « COMPILER DESIGN THEORY
LEWIS II, PH. M, ROSENKRANTZ, D.J. AND STEARNS.
R.E
LONDON: ADDISON-WESLEY PUBLISHING CO. 1975

3. «THE THEORY OF PARSING, TRANSLATION, AND COMPILING»
VOLUME I: PARSING
AHO AND ULIMAN
ENGLEWOOD CLIFFS, N.J.
PENPENTIC-HALL, INC., 1972

4. « COMPILER CONSTRUCTION, FOR DIGITAL COMPUTERS»
DAVID GRIES
JOHN-WILEY & SONS. INC. 1971

5. « SYNTAX OF PROGRAMMING LANGUAGES»
THEORY AND PRACTICE
ROLAND C. BACKHOUSE
PRENTIC-HALL INTERNATIONAL.

6. « INTRODUCTION TO FORMAL LANGUAGE THEORY»
MICHAEL A. HARRISON
ADDISON-WESLEY PUBLISHING CO. 1978