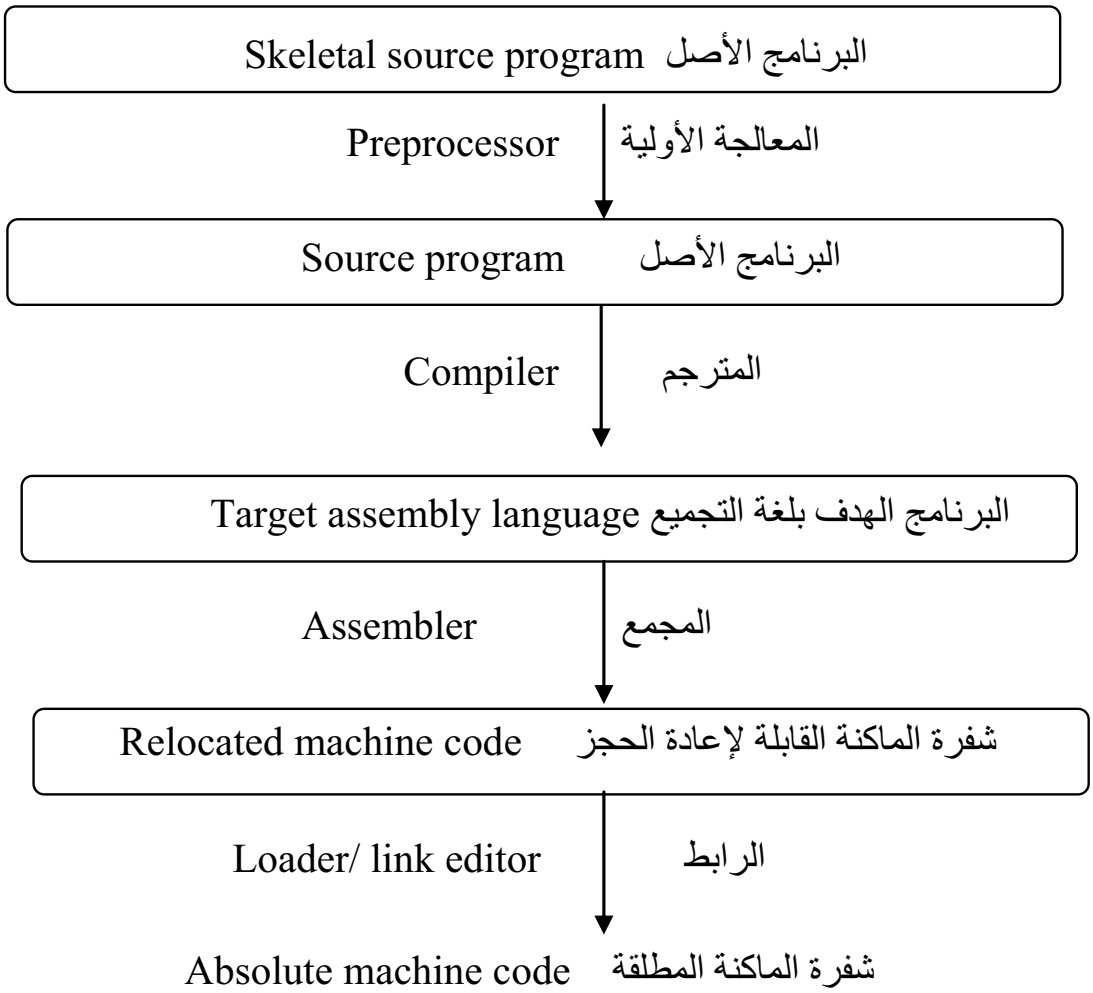
الأبجدية (Alphabet) هي مجموعه محددة من الرموز ويرمز لها بالرمز Σ وان هذه الرموز تسمى بالأحرف وان هذه الأحرف غير قابلة للتجزئة .



شكل (1.2) مراحل جعل البرنامج تنفيذي

اللغة (Language) هي مجموعة من الكلمات (السلاسل) المشكّلة من الرموز التي تنتمي إلى أبجدية مختارة مسبقا.

مثال ذلك إذا كانت الأبجدية { 0, 1 } = Σ فان واحدة من اللغات المشكّلة من هذه الأبجدية ستكون بالشكل التالي :

L = {0, 00, 000, 1, 11, 111, … }

الكلمة (word) هي سلسلة منتهية من الأحرف المشكلة من الأبجدية Σ وان هناك ما يسمى بالكلمة الفارغة (Empty word) والتي لا تتكون من أي حرف ويرمز لها بالرمز ε .

إن **العمليات** المعرّفة على اللغات (اللغة هي مجموعة سلاسل أو كلمات) هي كالآتي بفرض أن A و B مجموعتين:

- **الاتحاد** (Union) وتمثل على شكل :

$A \cup B = \{x \mid x \text{ in A or x is in B}\}$

- **التقاطع** (Intersection) وتمثل على شكل :

$A \cap B = \{x \mid x \text{ is in A and x is in B}\}$

- **الفرق** (Difference) وتمثل على شكل :

$A - B = \{x \mid x \text{ is in A but not in B}\}$

- **الإغلاق** (Closure) إن هذه العملية إذا كانت معرّفة على المجموعة A فإنها تشكل مجموعة كلمات منها الكلمة الخالية وجميع الكلمات المتشكلة من دمج مجموعة محددة من الكلمات الموجودة داخل المجموعة A ويمكن تمثيلها على الشكل التالي:

$A^* = A^0 \cup A^1 \cup A^2 \cup \ldots \cup A^n$

حيث تمثل المجموعة A^i حاصل دمج i من مجموعات الـ A كما يلي :

$$A^i = A \bullet A^{i-1}$$

حيث ان $A^0 = \{\varepsilon\}$ وان $A^1 = A$

وبالتالي فان A* هي مجموعة كل التركيبات الممكنة لعناصر المجموعة A.

فإذا كانت A={0,1} فان $A^0 = \{\varepsilon\}$ وان $A^1 = A$

و $A^2 = A \bullet A^1 = \{00,01,10,11\}$

وهكذا فان

$$A^* = \{\varepsilon,0,1,00,01,10,11,000,010,100,110,001,011,101,111,\ldots\}$$

- **مجموعة كل المجاميع** (Power Set) ويرمز لها بالرمز 2^A حيث أن :

$$2^A = \{P \mid P \text{ is a subset of } A\}$$

أي المجموعة التي تحتوي على كل المجموعات الجزئية للمجموعة A.

مثال : إذا كانت المجموعة A={0, 1, 2} فان

$$2^A = \{ \varphi,\{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\}\}$$

- **الدمج** (Concatenation) إذا كانت A و B مجموعتين فان عملية الدمج بين المجموعتين يمثل بـ AB={ab | a is in A and b is in B}

مثال 1.2 : إذا كانت A = { 0, 1 } وكانت B = { 1, 2 } فان

AB = {01, 02, 11, 12}

BA = {10, 11, 20, 21}

لذا فان $AB \neq BA$.

- **الضرب الكارتيزي** (Cartesian Product) إذا كانت A و B مجموعتين فان الضرب الكارتيزي بين المجموعتين يمثل بـ :

$$A \times B = \{(a, b) \mid a \text{ is in } A \text{ and } b \text{ is in } B\}$$

مثال : اذا كانت لدينا المجموعتين A = { 0, 1 } و B = { 1, 2 } فان

$$A \times B = \{(0, 1), (0, 2), (1, 1), (1, 2)\}$$

1.4 قواعد تركيب العبارة Phrase structure grammar (PSG)

هناك نوع من أنواع القواعد يسمى قواعد تركيب العبارة Phrase structure grammar (PSG) ويتكون من أربعة مكونات ويأخذ الشكل G=(N ,T ,P ,S) حيث أن

- N تمثل مجموعة محددة من الرموز الغير منتهية (nonterminals) والقابلة للاشتقاق التي تسمى بالأصناف القواعدية ويرمز لها بالرموز A,B,C,….

- أما الـ T تمثل مجموعة محددة من الرموز المنتهية (terminals)الغير قابلة للاشتقاق ويرمز لها بالرموز a,b,c,… حيث $N \cup T = V$ ويطلق على المجموعة V بالمفردات (vocabulary) وان N ∩ T = φ (المجموعة الخالية).
- P (production rules) مجموعة محددة من الأزواج المرتبة (α , β) أو على شكل (α → β) حيث أن كل من α و β ينتمون إلى V* وان α تتضمن على الأقل رمز من رموز المجموعة N .
- S رمز خاص ينتمي إلى مجموعة N يسمى رمز البداية (Start Symbol).

سيتم الاتفاق على نوع من الترميز كما يلي :

- α , β هي سلاسل (Strings) تنتمي الى V*.
- a ,b ,c هي سلاسل (Strings) تنتمي الى T* .
- A و B هي سلاسل (Strings) تنتمي إلى N* .

- إذا كانت A→B قاعدة إنتاج (Production Rule) في المجموعة P التي تم الإشارة إليها سابقا وكان αA→αB فيقال أن قاعدة الإنتاج A→B قد طبّقت على السلسلة αA للحصول على αB أو أن αB قد اشتقّت مباشرة من αA باستخدام القاعدة A→B.
- إذا كانت $\alpha \in V^*$ وبحيث $S \xrightarrow{*} \alpha$ (أي اشتقاق α من S بخطوة أو أكثر) فيقال عن α أنها نموذج جملي (sentential form) للقواعد G وبالتالي فان أي جملة في اللغة المقابلة للقواعد G هي نموذج جملي ينتمي إلى T* (هي السلاسل الغير قابلة للاشتقاق Terminal Strings التي يمكن اشتقاقها من رمز البداية S بخطوة أو أكثر) .
- اللغة المعرّفة على القواعد G هي كل الجمل المتولدة من القواعد G ويرمز للغة بالرمز L(G) وتعرف على شكل:

$$L(G) = \{x : x \in T^*, S \xrightarrow{*} x\}$$

مثال 1.3:

ليكن G=(N , T, P, S) قواعد بحيث أن N={S, A } وان T={a, b} وكان

P= {S→aAa ,

A→bAb ,

A→a}

هل أن الجملة abbabba$ ضمن اللغة المقابلة للقواعد أعلاه حيث أن الرمز $ هو علامة نهاية الجملة ؟

نعم هي كذلك لأن:

S→ **aAa** → a**bAb**a →ab**bAb**ba → abb**a**bba

حيث تم البدء برمز البداية ثم الاشتقاق أكثر من مرة وصولا إلى الجملة.

لو كان لدينا نوعين من القواعد G1 و G2 بحيث أن G1≠G2 وكانت اللغة المقابلة للقواعد G1 تساوي اللغة المقابلة للقواعد G2 أي L(G1)=L(G2) فان القواعد G1 تعتبر مكافئة للقواعد G2 (Equivalent).

ملاحظة : لو كانت لدينا السلسلة (string) abc فان :

Prefix = {ε ,a ,ab ,abc}

Suffix = { ε, c, bc, abc}

مثال 1.4 :

إذا كان(G=(N,T,P, S ، N={<integer>, <number>,<digit >,<fraction> }
، T={0,1,2,….,9,.} ، وأن S=<number>

P={<number> ::= <integer> <fraction>

<integer> ::= <integer> <digit> | <digit>

<fraction> ::= . <fraction> | < fraction> <digit> | <digit>

<digit> ::= 0|1| 2|…..|9 }

فهل أن الجملة 753.12 مقبولة ؟

نعم إنها مقبولة بسبب:

<number> ::= <integer> <fraction>

::= <integer> **.<fraction>**

::= < integer>.**<fraction> <digit>**

::= < integer>.<fraction> **2**

::= < integer>.**<digit>** 2

::= < integer>.**1**2

::= **< integer> <digit>** .12

::= < integer> **3** .12

::= **< integer> <digit>** 3 .12

::= < integer> **5**3 .12

::= **< digit >** 53 .12

::= 753 .12

حيث تم الاشتقاق من رمز البداية وصولا إلى الجملة.

1.5 هرم جومسكي Chomsky Hierarchy

يتضمن هذا الهرم أربعة أنواع من القواعد هي:

1. **Type 0**: في هذا النوع لا توجد شروط على القواعد وان قواعد الإنتاج تأخذ الشكل $\alpha \rightarrow \beta$ حيث أن كل من α و β ينتمون إلى V^* وأن $\alpha \neq \varepsilon$.

2. **Type 1** (Context Sensitive Grammar CSG):إن اللغة المتولدة من هذا النوع من القواعد تسمى (Context Sensitive Language) ومن صفات هذا النوع من القواعد إن قواعده تأخذ الشكل $\alpha \rightarrow \beta$ بحيث $|\alpha| <= |\beta|$ عدا القاعدة $S \rightarrow \varepsilon$ وان رمز البداية S لا يظهر في جهة اليمين من قواعد الإنتاج.

3. **Type 2** (Context Free Grammar CFG) من صفات هذا النوع أن قواعده تأخذ الشكل $NT \rightarrow \alpha$ حيث أن α تنتمي إلى V^* وان $|NT|=1$.

4. **Type 3** ويتضمن هذا الصنف نوعين من القواعد

a) Right Linear Grammar ويتألف من قواعد الإنتاج التي تأخذ الأشكال التالية:

$NT \rightarrow T\ NT$, $NT \rightarrow T$, $NT \rightarrow \in$

مثال 1.5:

$A \rightarrow bB$, $A \rightarrow b$, $A \rightarrow \varepsilon$

b) Left Linear Grammar ويتألف من قواعد الإنتاج التي تأخذ الأشكال :

$NT \rightarrow NT\ T$

$NT \rightarrow T$

$T \rightarrow \varepsilon$

مثال 1.6:

$A \rightarrow Bb$

$A \rightarrow b$

$A \rightarrow \varepsilon$

1.6 وصف أنواع القواعد

هناك عدة طرق لوصف القواعد:

- Transition diagram
- BNF
- EBNF
- Global meta language
- Syntax equations
- Regular Expressions (RE)

وان الطريقة الشائعة في وصف القواعد هي التعابير المنتظمة (Regular Expressions) التي سيتم شرحها في الفصل القادم .