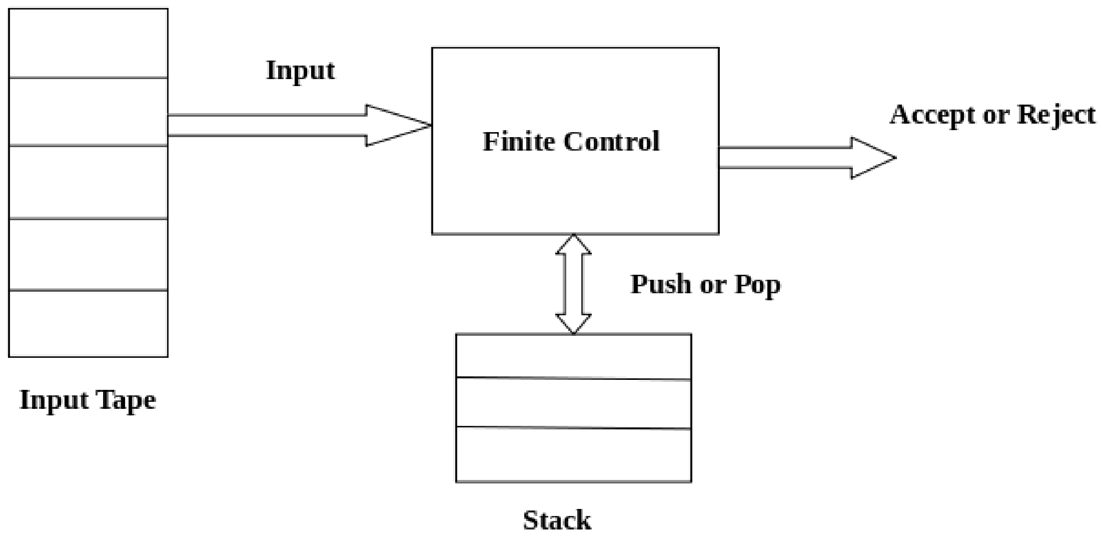
Pushdown Automata (PDA)

- Pushdown automata is a way to implement a context-free grammar (CFG) in the same way we design DFA for a regular grammar. A DFA can remember a finite amount of information, but a PDA can remember an infinite amount of information.
- Pushdown automata is simply an NFA augmented with an "external stack memory" ("Finite state machine" + "a stack"). The addition of stack is used to provide a last-in-first-out memory management capability to Pushdown automata. Pushdown automata can store an unbounded amount of information on the stack. It can access a limited amount of information on the stack. A PDA can push an element onto the top of the stack and pop off an element from the top of the stack. To read an element into the stack, the top elements must be popped off and are lost.
- A PDA is more powerful than FA. Any language which can be acceptable by FA can also be acceptable by PDA. PDA also accepts a class of language which even cannot be accepted by FA. Thus, PDA is much more superior to FA.



PDA Components:

Input tape: The input tape is divided in many cells or symbols. The input head is read-only and may only move from left to right, one symbol at a time.

Finite control: The finite control has some pointer which points the current symbol which is to be read.

Stack: The stack is a structure in which we can push and remove the items from one end only. It has an infinite size. In PDA, the stack is used to store the items temporarily.

Power of PDA

PDAs are more powerful than FSAs.

- $a^n b^n$, which cannot be recognized by an FSA, can easily be recognized by the PDA.
- Stack all a symbols and, for each b, pop an a off the stack.
- If the end of input is reached at the same time that the stack becomes empty, the string is accepted.

The languages accepted by PDAs are equivalent to the context-free languages.

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

Formal definition of PDA:

The PDA can be formally described as a collection of 7-tuple/components (Q, ∑, Γ, q_0, z_0, F, δ):

Q: the finite set of states.
∑: the finite set of symbols which is called the input alphabet.
Γ: a stack symbol which can be pushed and popped from the stack.
q_0: the initial/start state, $q_0 \in Q$.
z_0: a starting stack symbol, $Z \in \Gamma$, the PDF's stack consists this symbol and nothing else.
F: a set of final/acceptance states, $F \subseteq Q$.
δ: mapping/transition function which is used for moving from current state to next state.

In a given state, PDA will read input symbol and stack symbol (top of the stack) and move to a new state and change the symbol of stack.

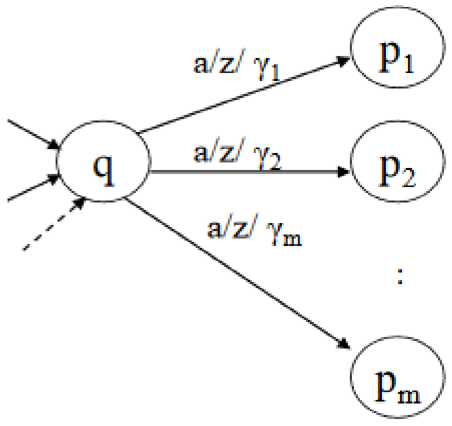
Instantaneous Description (ID)

ID is an informal notation of how a PDA computes an input string and make a decision that string is accepted or rejected.

An instantaneous description is represented by a triple (q, w, α), where:

q describes the current state.
w describes the remaining input (unconsumed input).
α describes the stack contents, top at the left.

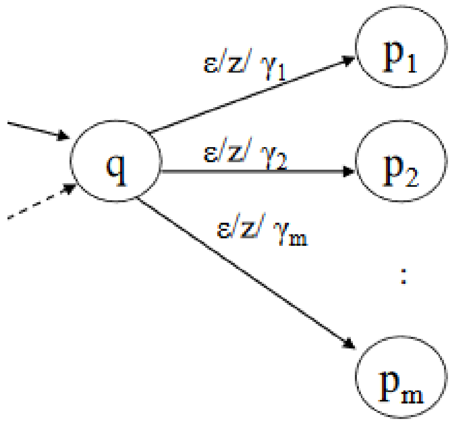
Types of PDA transitions:



1st Type:

$\delta(q, a, z) = \{(p_1,\gamma_1), (p_2,\gamma_2),\ldots, (p_m, \gamma_m)\}$

- Current state is q
- Current input symbol is a
- Symbol currently on top of the stack z
- Move to state p_i from q
- Replace z with γ_i on the stack (leftmost symbol on top)
- Move the input head to the next input symbol



2nd Type:

$\delta(q, \varepsilon, z) = \{(p_1,\gamma_1), (p_2,\gamma_2),\ldots, (p_m, \gamma_m)\}$

- Current state is q
- Current input symbol is not considered
- Symbol currently on top of the stack z
- Move to state p_i from q
- Replace z with γ_i on the stack (leftmost symbol on top)
- **No input symbol is read**

Example 1.

Construct a PDA automata for language $L = \{0^n 1^n \mid n \geq 0\}$

$M = (\{q_0, q_1\}, \{0, 1\}, \{z_0, 0\}, \delta, q_0, z_0, q_f)$ // Here $\{q_0, q_1\}$ states, {0, 1} alphabet, $\{z_0, 0\}$ input to stack, q_0 is the initial state, z_0 represents that stack is empty, and q_f is the final state.

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

The δ can be:

1. $\delta(q_0, 0, z_0) = \{(q_0, \overrightarrow{0z_0})\}$ // stack order: 0 on top, then z_0 below
2. $\delta(q_0, 1, z_0) = q_f$ // illegal, string rejected. When will it happen?
3. $\delta(q_0, 0, 0) = \{(q_0, 00)\}$
4. $\delta(q_0, 1, 0) = \{(q_1, \varepsilon)\}$ // ε indicates pop operation
5. $\delta(q_1, 1, 0) = \{(q_1, \varepsilon)\}$
6. $\delta(q_1, \varepsilon, z_0) = \{(q_f, z_0)\}$ // if ε read & stack hits bottom, accept.
7. $\delta(q_1, \varepsilon, 0) = q_f$ // illegal, string rejected. When will it happen?
8. $\delta(q_0, \varepsilon, z_0) = \{(q_f, z_0)\}$ // n=0, accept. When will it happen?

- **Goal:** (acceptance)
  - Read the entire input string
  - Terminate with an empty stack
- Informally, a string is accepted if there exists a computation that uses up all the input and leaves the stack empty.

Turnstile Notation:

$\vdash$ sign describes the turnstile notation and represents one move.

$\vdash^*$ sign describes a sequence of moves.

For example,

$(p, b, T) \vdash (q, w, \alpha)$

This implies that while taking a transition from state p to state q, the input symbol 'b' is consumed, and the top of the stack 'T' is represented/replaced by a new string α.

Example 2.

Design a PDA for accepting a language $\{a^n b^{2n} \mid n >= 1\}$.

Solution.

In this language, n number of a's should be followed by 2n number of b's. Hence, we will apply very simple logic, and that is if we read single 'a', we will push two a's onto the stack. As soon as we read 'b' then for every single 'b' only one 'a' should get popped from the stack.

The ID can be constructed as follows:

$\delta(q_0, a, z_0) = (q_0, aaz_0)$

$\delta(q_0, a, a) = (q_0, aaa)$

Now when we read b, we will change the state from q_0 to q_1 and start popping corresponding 'a'. Hence,

$\delta(q_0, b, a) = (q_1, \varepsilon)$

Thus, this process of popping 'b' will be repeated until all the symbols are read. Note that popping action occurs in state q_1 only.

$\delta(q_1, b, a) = (q_1, \varepsilon)$

After reading all b's, all the corresponding a's should get popped. Hence when we read ε as input symbol then there should be nothing in the stack. Hence the move will be:

$\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$

Where,

PDA = $(\{q_0, q_1\}, \{a, b\}, \{a, z_0\}, \delta, q_0, z_0, q_f)$

We can summarize the ID as:

1. $\delta(q_0, a, z_0) = (q_0, aaz_0)$
2. $\delta(q_0, a, a) = (q_0, aaa)$
3. $\delta(q_0, b, a) = (q_1, \varepsilon)$
4. $\delta(q_1, b, a) = (q_1, \varepsilon)$
5. $\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$

Now we will simulate this PDA for the input string "aaabbbbbb".

$\delta(q_0, aaabbbbbb, z_0)$

$\vdash \delta(q_0, aabbbbbb, aaz_0)$

$\vdash \delta(q_0, abbbbbb, aaaaz_0)$

$\vdash \delta(q_0, bbbbbb, aaaaaaz_0)$

$\vdash \delta(q_1, bbbbb, aaaaaz_0)$

$\vdash \delta(q_1, bbbb, aaaaz_0)$

$\vdash \delta(q_1, bbb, aaaz_0)$

$\vdash \delta(q_1, bb, aaz_0)$

$\vdash \delta(q_1, b, az_0)$

$\vdash \delta(q_1, \varepsilon, z_0)$

$\vdash \delta(q_f, z_0)$ **ACCEPT**

Example 3.

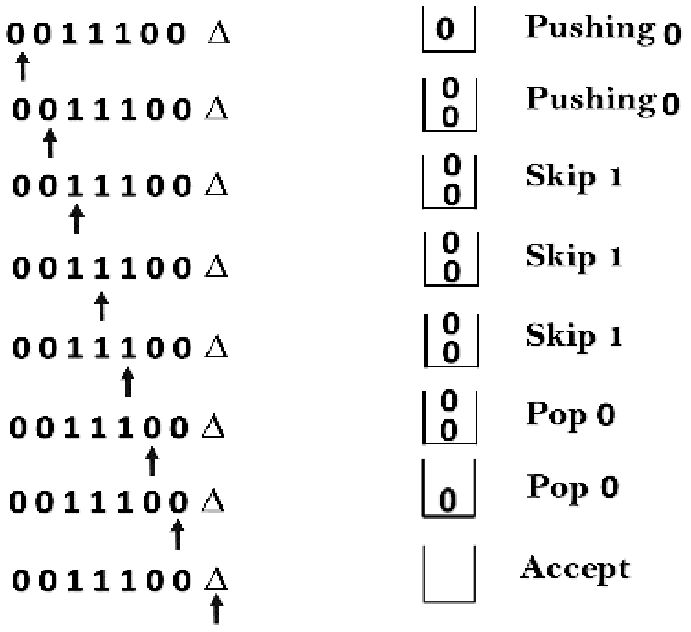
Design a PDA for accepting a language $\{0^n 1^m 0^n \mid m, n >= 1\}$.

Solution.

In this PDA, n number of 0's are followed by any number of 1's followed n number of 0's. Hence the logic for design of such PDA will be as follows:

Push all 0's onto the stack on encountering first 0's. Then if we read 1, just do nothing. Then read 0, and on each read of 0, pop one 0 from the stack. **For instance,**

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.



This scenario can be written in the ID form as:

1. $\delta(q_0, 0, z_0) = \delta(q_0, 0z_0)$
2. $\delta(q_0, 0, 0) = \delta(q_0, 00)$
3. $\delta(q_0, 1, 0) = \delta(q_1, 0)$
4. $\delta(q_1, 1, 0) = \delta(q_1, 0)$
5. $\delta(q_1, 0, 0) = \delta(q_2, \varepsilon)$
6. $\delta(q_2, 0, 0) = \delta(q_2, \varepsilon)$
7. $\delta(q_2, \varepsilon, z_0) = \delta(q_f, z_0)$ **ACCEPT**

Now we will simulate this PDA for the input string "0011100".

$\delta(q_0, 0011100, z_0)$
$\vdash \delta(q_0, 011100, 0z_0)$
$\vdash \delta(q_0, 11100, 00z_0)$
$\vdash \delta(q_1, 1100, 00z_0)$
$\vdash \delta(q_1, 100, 00z_0)$
$\vdash \delta(q_1, 00, 00z_0)$
$\vdash \delta(q_2, 0, 0z_0)$
$\vdash \delta(q_2, \varepsilon, z_0)$
$\vdash \delta(q_f, z_0)$ **ACCEPT**

Example 4.

Construct a PDA automata for the language $L = \{x \mid x \in ww^R$ and w in $\{a, b\}^*\}$.

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

Solution.

Suppose the language consists of string L = {aa, bb, abba, baab, aabbaa, bbaabb, abaaba, ...]. The string can be odd palindrome or even palindrome. The language, Lww^R, is the even-length palindromes over alphabet {0,1} i.e.,, length |x| is even. Lww^R is a Context-Free Language (CFL) generated by the grammar: $S \rightarrow aSa \mid bSb \mid \varepsilon$.

The logic for constructing PDA is that we will push a symbol onto the stack till half of the string then we will read each symbol and then perform the pop operation. We will compare to see whether the symbol which is popped is similar to the symbol which is read. Whether we reach to end of the input, we expect the stack to be empty.

This PDA is a non-deterministic PDA because finding the mid for the given string and reading the string from left and matching it with from right (reverse) direction leads to non-deterministic moves.

$M = (\{q_0, q_1\}, \{a, b\}, \{z_0, a\}, \delta, q_0, z_0, q_f)$

The δ can be:

1. $\delta(q_0, a, z_0) = (q_0, az_0)$
2. $\delta(q_0, b, z_0) = (q_0, bz_0)$
3. $\delta(q_0, a, a) = (q_0, aa)$
4. $\delta(q_0, a, b) = (q_0, ab)$
5. $\delta(q_0, b, a) = (q_0, ba)$
6. $\delta(q_0, b, b) = (q_0, bb)$

Push the symbols onto the stack.

7. $\delta(q_0, a, a) = (q_1, \varepsilon)$
8. $\delta(q_0, b, b) = (q_1, \varepsilon)$
9. $\delta(q_1, a, a) = (q_1, \varepsilon)$
10. $\delta(q_1, b, b) = (q_1, \varepsilon)$

Popping the symbols on reading the same kind of symbol.

11. $\delta(q_0, \varepsilon, z_0) = (q_f, z_0)$
12. $\delta(q_1, \varepsilon, z_0) = (q_f, z_0)$

Pop the final stack symbol off at the end of a computation.

Note:

Rules 3, 7 and 6, 8 are non-deterministic: two options each.

Simulation of "abaaba":

$\delta(q_0, abaaba, z_0)$ Apply rule 1

$\vdash \delta(q_0, baaba, az_0)$ Apply rule 5

$\vdash \delta(q_0, aaba, baz_0)$ Apply rule 4

$\vdash \delta(q_0, aba, abaz_0)$ Apply rule 7

$\vdash \delta(q_0, ba, baz_0)$ Apply rule 8

$\vdash \delta(q_1, a, az_0)$ Apply rule 9

$\vdash \delta(q_1, \varepsilon, z_0)$ Apply rule 12

$\vdash \delta(q_1, \varepsilon)$ **ACCEPT**

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

Example 5.
Construct a PDA that accepts the language L over {0, 1} by empty stack which accepts all the string of 0's and 1's in which a number of 0's are twice of number of 1's.

Solution.
There are two parts for designing this PDA:
- If 1 comes before any 0's.
- If 0 comes before any 1's.

We are going to design the first part i.e.,, 1 comes before 0's. The logic is that read single 1 and push two 1's onto the stack. Thereafter on reading two 0's, POP two 1's from the stack.
The δ can be:
$\delta(q_0, 1, z_0) = (q_0, 11z_0)$
$\delta(q_0, 0, 1) = (q_0, \varepsilon)$

Now, consider the second part i.e.,, if 0 comes before 1's. The logic is that read first 0, push it onto the stack and change state from q_0 to q_1. [Note that state q_1 indicates that first 0 is read and still second 0 has yet to read].

Being in q_1, if 1 is encountered then POP 0. Being in q1, if 0 is read then simply read that second 0 and move ahead.
The δ will be:
$\delta(q_0, 0, z_0) = (q_0, 0z_0)$
$\delta(q_0, 0, 0) = (q_0, 00)$
$\delta(q_0, 1, 0) = (q_1, \varepsilon)$ (indicate that one 0 and one 1 is already read, so simply read the second 0)
$\delta(q_1, \varepsilon, 0) = (q_0, \varepsilon)$

Now, summarize the complete PDA for given L is:
1. $\delta(q_0, 0, z_0) = (q_0, 0z_0)$
2. $\delta(q_0, 0, 0) = (q_0, 00)$
3. $\delta(q_0, 1, z_0) = (q_0, 11z_0)$
4. $\delta(q_0, 1, 1) = (q_0, 111)$
5. $\delta(q_0, 0, 1) = (q_0, \varepsilon)$
6. $\delta(q_0, 1, 0) = (q_1, \varepsilon)$
7. $\delta(q_1, \varepsilon, 0) = (q_0, \varepsilon)$
8. $\delta(q_1, \varepsilon, z_0) = (q_0, 1z_0)$
9. $\delta(q_0, \varepsilon, z_0) = (q_f, z_0)$ **ACCEPT**

HW. simulate this PDA for the input strings: "000011", "110000", "100001", "001001", "100010", "001010", "011000".

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

Example 6.
Construct a PDA for the language L = $\{x \mid x \in \{a, b, c\}^* \mid$ where $|x|_c = 2*|x|_b + 3*|x|a$.

Solution.
The δ can be:

1. $\delta(q_0, a, z_0) = (q_0, ---z_0)$
2. $\delta(q_0, b, z_0) = (q_0, --z_0)$
3. $\delta(q_0, c, z_0) = (q_0, +z_0)$
4. $\delta(q_0, a, -) = (q_0, ----)$
5. $\delta(q_0, b, -) = (q_0, ---)$
6. $\delta(q_0, c, +) = (q_0, ++)$
7. $\delta(q_0, c, -) = (q_0, \varepsilon)$
8. $\delta(q_0, a, +) = (q_1, \varepsilon)$
9. $\delta(q_1, \varepsilon, +) = (q_2, \varepsilon)$
10. $\delta(q_2, \varepsilon, +) = (q_0, \varepsilon)$
11. $\delta(q_1, \varepsilon, z_0) = (q_0, --z_0)$
12. $\delta(q_2, \varepsilon, z_0) = (q_0, -z_0)$
13. $\delta(q_0, b, +) = (q_4, \varepsilon)$
14. $\delta(q_4, \varepsilon, +) = (q_0, \varepsilon)$
15. $\delta(q_4, \varepsilon, z_0) = (q_0, -z_0)$
16. $\delta(q_0, \varepsilon, z_0) = (q_f, z_0)$

HW. Simulate this PDA for the input strings: "abccccc", "acccbcc", "bccaccc", "cccaccb", "ccacccb", "cacccbccacc", "cacccbbccc", "cbaabcccccccccc", "baabacccccccccccccc", "ccbabacccccccccc".

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.

HW 1. *Construct a PDA for language* $L = \{a^n b^n \mid n \geq 1\}$

HW 2. *Construct a PDA for language* $L = \{a^i b^j c^k \mid i+j=k;\ i, j, k \geq 1\}$

HW 3. *Construct a PDA for language* $L = \{a^m b^{(2m)} \mid m \geq 1\}$

HW 4. *Construct a PDA for language* $L = \{a^n b^m c^n \mid n, m \geq 1\}$

HW 5. *Construct a PDA for language* $L = \{a^n b\, a^{2n} \mid n \geq 0\}$

HW 6. *Construct a PDA for language* $L = \{0^n 1^m \mid n, m \geq 1 \text{ and } m > n+2\}$

HW 7. *Construct a PDA for language* $L = \{0^n 1^m 2^m 3^n \mid m, n \geq 0\}$

HW 8. *Construct a PDA for language* $L = \{a^m b^i c^m d^k \mid m, i, k \geq 0\}$

HW 9. *Construct a PDA for language* $L = \{a^m b^m c^n d^n \mid m, n \geq 0\}$

HW 10. *Construct a PDA for language* $L = \{a^i b^j c^k d^l \mid i=k;\ j=l\}$

HW 11. *Construct a PDA for language* $L = \{a^m b^n c^p d^q \mid m+n = p+q;\ m, n, p, q \geq 1\}$

HW 12. *Construct a PDA automata for language* $L = \{0^m 1^{(n+m)} 2^n \mid m, n \geq 0\}$

HW 13. *Construct a PDA automata for language* $L = \{0^{(n+m)} 1^m 2^n \mid m, n \geq 0\}$

HW 14. *Construct a PDA automata for language* $L = \{0^n 1^m 2^{(n+m)} \mid m, n \geq 0\}$

HW 15. *Construct a PDA automata for language* $L = \{a^{(2*m)} c^{(4*n)} d^n b^m \mid m, n \geq 0\}$

HW 16. *Construct a PDA for language* $L = \{w\, c\, w' \mid w \text{ belongs to } \{a, b\}^*\}$ *where* $|w| \neq 0$ *and* w' *is the reverse of* w. For example: if $w = ABC$ *then* $w' = CBA$

HW 17. *Construct a PDA for language* $L = \{w\, w' \mid w \text{ belongs to } \{a, b\}^*\}$ *where* $|w| \neq 0$ *and* w' *is the reverse of* w.

HW 18. *Construct a PDA for language* $L = \{w \in \{a, b, c\}^*$ *where* $|w|_a + |w|_b = |w|_c$.

HW 19. *Construct a PDA for the language* $L = \{w \in \{a, b\}^* \mid$ *w contains equal no. of a's and b's.*

HW 20. *Construct a PDA for language* $L = \{0^n\, 2\, 3\, 1^n \mid n \geq 0\}$

HW 21. *Construct a PDA for language* $L = \{a^{n+3} b^{n+2} \mid n > 0\}$

HW 22. *Construct a PDA for language* $L = \{a^{2n} b^n \mid n \geq 0\}$

HW 23. *Construct a PDA for language* $L = \{a^{2n+4} b^{n+1} \mid n \geq 0\}$

HW 24. *Construct a PDA for language* $L = \{w\, c\, d\, w' \mid w \text{ belongs to } \{a, b\}^*\}$ *where* $|w| \neq 0$ *and* w' *is the reverse of* w.

HW 25. *Construct a PDA for language* $L = \{0^i 1^j \mid i \geq 0 \text{ and } i \leq j \leq 2i\}$

(*) أستاذ مساعد، قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الموصل.