تَوْصِيفُ وحَوْسَبَةُ قَواعِدِ التَّصْغِيرِ الصَّرْفِيَّة – مُقَارَبَة بَرْمَجِيَّة بِلُغَة جَافَاسكريبت

Characterization and computerization of Altasghir rules – JavaScript programming approach

بخيرة الحسين[1]*، أ.د سعاد بسناسي[2]

[1] عبد الحميد بن باديس مستغانم (الجزائر)، bekhiraelhossien@gmail.com

[2]جامعة وهران 1 أحمد بن بلة (الجزائر)، Besnacisouad@yahoo.fr

تاريخ الاستلام:2022-10-25 تاريخ القبول: 2022-11-04 تاريخ النشر:2022-12-27

مُلخَّص البَحث

عَرَفَتْ اللغة العربية عِدَّة محاولات في مجال تطوير صياغة الأنماط اللغوية الخاصة بها إلى أنماط رياضية حاسوبية، وذلك خدمة للغة العربية في مجالات الحَوْسَبَة والرقمنة؛ ولهذا جاءت دراستنا كمحاولة لتوصيف بعض قواعد التَّصغير الصَّرفية، وصياغتها صياغةً رياضيَّة منطقيَّة ممَّا يُسَهِّل عمليَّة بَرْمَجَتِهَا. وقد اخترنا لهذه العملية لغة (جافاسكريبت) (*JavaScript*) المشهورة والقوية في مجال الويب.

كلمات مفتاحية: التوصيف اللساني، الصياغة الرياضية الحاسوبية، باب التصغير، لغة جافاسكريبت.

Abstract:

The Arabic language has known several attempts in the field of formulating its own linguistic patterns into computer mathematical patterns, in order to serve the Arabic language in the fields of computing and digitization, and this is why our study came as an attempt to describe some of Altasghir rules and formulate them in a logical mathematical form, which facilitates the process of programming because JavaScript is a very popular and powerful language in the web.

Keywords: Linguistic characterization; Computational mathematical formulation; Bab Altasghir; JavaScript.

* **المؤلف المرسل: بخيرة الحسين**

1. مقدمة:

يعتبر التَّوصيف عملية مهمَّة تساعد بشكل كبير في فهم الظواهر اللغوية، وهو اللَّبِنَة الأولى في الانتقال من المستوى اللساني إلى المستوى الحاسوبي، وعليه فأيُّ محاولة توصيفية جادة لظواهر اللغة العربية هي في حقيقتها مساهمة فعلية للارتقاء بالواقع اللساني الحاسوبي. واللغة العربية في هذا العصر الذي نعيشه بأمَسِّ الحاجة إلى من يضعها في سكة الطريق الرقمي، ولهذا بدأت المحاولات الجادة في رقمنة وتوصيف وحوسبة فروع اللغة العربية سواء من طرف الباحثين العرب أنفسهم أم من طرف الشركات الكبرى العالمية كغوغل وفيس بوك وغيرها...، والتي سَعَتْ إلى احتواء جمهور عربي كبير توحده لغة واحدة، و قد استطاعت إلى حدٍّ ما هذه المحاولات تذليل العقبات وتقريب الهوَّة بين ما كان مُترسِّخا في الذِّهن العربي من ادعاء أنَّ اللغة العربية لغة عقيمة تستعصي على كل خطَّة رقميَّة عصريَّة، وبين قوَّة اللغة العربية ومنطِقِيَّتِها بين لغات العالم. حيث اتَّضَح للعرب أنفسهم بعد هذه المحاولات واقعا ناصِعًا في مُرُونة لغتهم وأدائها الحاسوبي؛ غير أنَّ هذه التجارب بقيت حِكْرًا على أصحابها ولم تحظ المناهج والخوارزميات التي وظفوها بالمشاركة الكافية مع جمهور الباحثين المهتمين بتطوير حوسبة اللغة العربية، ناهيك عن المحاولات التوصيفية العديدة لظواهر اللغة العربية التي قام بها الكثير من الباحثين والتي بقيت أغلبها مجردة من التطبيق الحاسوبي، وهذا ما جعلها حبيسة المجال النظري فقط، لا تتجاوز حدود كونها مجرد مفاهيم نائية عن المفهوم الأساسي لواقع اللسانيات الحاسوبية.

وعلى هذا الأساس جاءت دراستنا الموسومة بـ: «تَوْصِيفُ وحَوْسَبَةُ قَواعِدِ التَّصْغِيرِ الصَّرْفِيَّة – مُقَارَبَة بَرْمَجِيَّة بِلُغَة جَافَاسكريبت» كمحاولة توصيفية تطبيقية بالدرجة الأولى، تتدرج في مسالك هذا البحث بكل دقائقه حتى تصل إلى بناء برنامج حاسوبي للتَّصغير الصرفي بلغة برمجية مشهورة وهي لغة (جافاسكريبت)، وبذلك تكون أداة فعالة في هذا الميدان اللساني الحاسوبي.

وتأتي أهمية هذا البحث من كونه دراسة جمعت بين ما هو نظري وتطبيقي، حيث لا خروج عن القصد المرجو من هذا البحث فيما لا طائل منه، وكون هذا النوع من البحوث مطلوب عند الباحثين المهتمين بمجال دراسات اللسانيات الحاسوبية.

ولقد كان من دواعي اختيارنا لهذا البحث عدَّة أسباب هي المسوغات الفعلية لقيامه من الأساس وهي كالآتي:

- تعزيز الدراسات التوصيفيَّة في المجال اللِّسانيِّ الحاسوبيِّ.
- الإجابة عن بعض الإشكاليات المطروحة في الدرس التَّوصيفيِّ اللُّغَويِّ العربيِّ.
- تقريب لغات البرمجة الحديثة من الباحث اللسانيِّ حتَّى يَتَسَنَّى له معرفة طرق الاستفادة منها في إجراء مقاربات لغوية.

ولهذا البحث عدَّة أهداف يمكن حصرها فيما يلي:

- إعداد منهجيَّة التَّوصيف المناسب للبرمجيَّة الحاسوبيَّة.
- توظيف المفاهيم المنطقية والمفاهيم الرياضية المصاحبة للغة العربية.
- بيان طرق الحصر والإحصاء للأوزان الصرفية بمقاربات برمجية.
- التطبيق الفعليُّ للإسقاط اللغوي على القواعد البرمجيَّة.
- الحل البرمجيُّ لبعض مشاكل اللغة العربيَّة كالتَّشكيل وقياس بنية الكلمة.

وتحقيقا لهذا المسعى وضعنا جملة من التساؤلات التي سوف تساهم في بيان معالم هذا البحث:

- إلى أيِّ مدًى يمكن توصيف وحوسبة قواعد التَّصغير الصَّرفية؟
- ما هي أهَمِّ المشاكل المنطقيَّة والبرمجيَّة التي تواجه الباحث في مثل هذه البحوث اللغوية؟
- هل قواعد اللُّغة العربيَّة منطقيَّة إلى هذا الحدِّ الحاسوبيِّ أم أنَّها غير ذلك؟
- إلى أيِّ مدى يمكن توظيف لغة (جافاسكريبت) في الدَّرس الصَّرفي العربي؟

وقد اتَّبعنا في دراستنا هذه المنهج الوصفيَّ التَّحليليَّ فِيما يَخُصُّ عَمليَّة التَّوصيف لِظاهرة التَّصغِير الصَّرفيِّ، واتَّبعنا المنهج الرياضيَّ البرمجيَّ المنطقيَّ في بناء البرنامج والخوارزميَّات المناسبة للتَّوصيف بلغة (جافاسكريبت).

2. حدود توصيف قواعد التَّصغير في هذه الدِّراسة:

يعتبر باب التَّصغير جزءا مهِمًّا من أجزاء الدرس اللغوي الصرفي، وهو في حقيقته وضع من أوضاع اللغة العربية مثله مثل الأبواب الأخرى في فن الصَّرف، وعند التأمل في قواعده وضوابطه جيدا نَجِدُ أنَّهُ يَنْمَازُ بقواعد قياسية مطَّرِدَة ومعانٍ ودلالات تدلُّ عليها صِيَغُهُ الصَّرفية؛ فقاعدة تصغير الثلاثي مثلا تمشي وفق قياس مُطَّرد وهو تصغير ما كان على وزن «فَعَل »إلى وزن «فُعَيْل»، ويتخلل هذا القياس كثير من تقلبات الأوضاع الصَّرفية، كوجود تاء التأنيث في آخرِ بنية الثلاثي مثل كلمة «بقرة» مثلا، ووجود بعض حروف العلِّة في أصل الكلمة المصغرة أيضا وغيرها مما سوف يأتي ذكره في موضعه.

وبناء عليه يمكننا أن نحدد قواعد باب التَّصغير إجمالا دون الاستفاضة أو ذكر الخلاف الحاصل فيها بين المدارس النَّحوية والصَّرفيَّة، واستنادا على ما ورد ذكره في جامع الدروس العربية لمصطفى الغلاييني يتألف باب التَّصغير من هذه المحاور الآتية:

- مفهوم التصغير وماهيته.
- شروط التَّصغير وما شذَّ من التصغير عن هذه الشروط.
- فائدة التَّصغير ودلالة صِيَغِه الصَّرفية كالتقليل والتحقير والتقريب مثلا.
- حكم ما بعد ياء التَّصغير.
- أوزان التَّصغير في الاسم الثلاثي والرباعي والخماسي، والخماسي الذي كل أحرفه الخمسة أصلية، وما بلغت أحْرُفه بالزيادة أكثر من أربعة، وما كان يحتوي في بنيته على ألف التأنيث المقصورة أو تاء التأنيث أو الألف الممدودة وغيرها من العوارض اللغوية.
- تصغير ما ثانيه حرف علة. تصغير ما ثالثه حرف علة.
- تصغير ما رابعه حرف علَّة.
- تصغير ما حذف عنه شيء.
- تصغير الثنائي الوضع.
- تصغير المؤنث.

- تصغير العلم المركب.
- تصغير الجمع.
- تصغير التَّرخيم.
- شواذ التَّصغير[1].

وبعد اطلاعنا وفحصنا لكل هذه المحاور الواردة في باب التَّصغير، طرَحنَا هذا السُّؤال: هل يمكننا أن نخضعها جميعها لعمليَّة التَّوصيف القَابل للحَوسبة؟

وللإجابة عن هذا السؤال إجابة مبنية على تعليل واضح وصريح، يُتَطَلّب منَّا قبل كلِّ شيء القيام بعملية تحليلية دقيقة لبعض المحاور التي تقدمت؛ ويمكننا البدء بمحور شروط التَّصغير مثلا، فمن شروط التَّصغير أن يكون الاسم معربا، قابلا للتَّصغير، ومعنى هذا إيجاد توصيف كامل منطقي يمكن معالجته على مستوى الحاسب الآلي بلغة برمجية قادرة على التكيف مع معطياته، وتوصيف الاسم المعرب وحده وتحويره إلى منطق رياضي قابل للبرمجة أمر عسير جدا، إن لم نقل هو في عداد الاستحالة. فتوصيف الاسم المعرب يتضمن توصيف الاسم المبني لأنَّ القسمة ثنائية فالاسم إمَّا معرب أو مبني، فعلى أي أساس صار الاسم معربا ومبنيا؟ وهل العلة في إعرابه قياسية أم سماعية؟ فإن كانت قياسية فهل يمكن توصيفها من دون استدعاء توصيف شامل لعناصر أخرى خارجة عن باب الاسم المعرب؟وإن كانت سماعية فنحن نعلم أنَّ مَرَدَّها إلى استدعاء قاعدة بيانات معجمية تميز كل ما هو سماعي وتعطي الحكم مباشرة من كتب النحو والصَّرف، وهذا أمر يستدعي تضافر جهود متواصلة من البحث لفريق كبير من الباحثين في الموضوع، وذلك لشيوع كثرة الخلاف وعوائق الاشتقاق في اللغة العربية.

والسؤال نفسه يطرح بالنسبة للاسم المبني أيضا، فالمبني كما هو معلوم في اللغة سبعة أنواع: بين اسم إشارة واسم موصول واسم فعل وغير ذلك...، وتوصيف الاسم المبني وحده يتطلب توصيف كل هذه الأنواع بما فيها من أصل سماعي مما يعود بنا مرَّة أخرى إلى قاعدة البيانات المعجمية. وهبْ أنَّنا بلغنا هذا الحدَّ من التوصيف في السَّماعي فعلى أي مدرسة نحكم في الخلاف النَّحوي والصَّرفي ونحن ندرك أن الخلاف واسع وعريض بين الكوفيين والبصريين؟ وهل هو مستساغ إخبار المستخدم للآلة أنَّ هذا الاسم

معرب أو مبني لأنَّ المدرسة الفلانية حكمت بذلك؟ وأمر آخر يذهب بنا بعيدا عن كلِّ هذا، فهب أنَّنا وجدنا الحل الأنسب، فهل نحن قادرون على بناء خوارزمية تفتش في قاعدة البيانات المعجمية الضخمة جدا – والمؤلفة من آلاف الدواوين على اختلاف مشاربها ومدارسها– بين المعاني الواردة في الجذر الفلاني الواحد مثلا؟ لاسيما وأنَّ أوزان الصَّرف العربي تحمل معان خاصة في وضعها اللغوي كصيغ المبالغة وجموع القلة والكثرة مثلا، فأنَّى لنَا إذِن منْ قُدْرَةٍ على توصيفها وبناء خوارزميات للكشف عنها؟

كلُّ ما تقدَّم من هذه الإشكالات العويصة يدور حول شرط واحد فقط من شروط التصغير، ألا وهو: أن يكون الاسم معربا، فكيف بنا لو بحثنا في محاولة توصيف كل الشروط الأخرى؟ ولو تابعنا الاستقصاء إلى محاور أخرى كمحور دلالة صيغ التصغير وكذلك محور شواذ التصغير فإننا نجد أن أمْر توصيف الصيغة سماعي ومعنوي لا دَخْل للقياس فيه، أي أنه راجع لاستدعاء قاعدة بيانات معجمية كما تقدم القول في ذلك.

ومجمل القول فإنَّ توصيف باب التَّصغير كاملا أمرٌ ليس في متناول هذه الدراسة المختصرة، وإنَّما المجال المغلق الذي سوف تقوم دراستنا عليه في عملية التوصيف هو هذا النَّص من قاعدة التَّصغير لا غير:

"الاسم الثلاثي: يصغر على صيغة (فُعَيْل)، وذلك بأنْ نَضُمَّ الحرف الأول، ونفتح الحرف الثاني، ثم نزيد بعده ياء ساكنة هي التي تسمى ياء التصغير، ثم يأتي الحرف الثالث دون تغيير، فنقول: رَجُل ورُجَيْل، نَهْر ونُهَيْر، جَبَل وجُبَيْل، وَلَدٌ ووُلَيْد فإن كان الاسم الثلاثي بعده تاء تأنيث فإنَّها لا تؤثر على هذه العملية، فنقول: بَقَرَة وبُقَيْرَةٌ، شَجَرَةٌ وشُجَيْرَةٌ"[2].

"الاسم الرُّباعي: يصغر على صيغة (فُعَيْعِل) أي بأن نَضُمَّ الحرف الأول، ونفتح الحرف الثاني، ثم نزيد ياء التَّصغير السَّاكنة، ثم نكسر الحرف الذي بعدها، فنقول: جَعْفَر وجُعَيْفِر، مَسْجِدٌ ومُسَيْجِد، بُنْدُق وبُنَيْدِق مَنْزِلٌ ومُنَيْزِل"[3].

فهذا هو النَّص الذي سوف تُعْنى دراستنا بتوصيفه وحوسَبَتِه بلغة (جافاسكريبت)، وكما نلاحظ في هذا النَّص غياب توصيف الثنائي الوضع، وذلك لأنَّ قياس تصغيره على وزن قياس الثلاثي «فُعَيْل»، أي أنَّ الحرف الثالث محذوف وهو تكرار للحرف الثاني فقط مثل: «هَلْ» و«هُلَيْل»، فمن هذه العلَّة تركنا

توصيف ثنائي الوضع لأنَّ توصيفه قبل الثلاثي غير منطقي أبدا، وتركنا أيضا توصيف ما كان أحد أبنيته حرف علة لأنَّ هذا النوع من الاسم يرجع إلى أصله في التصغير،والرجوع إلى الأصل في الصَّرف سماعي وتوقيفي لا دخل للقياس فيه على الغالب، وهذا ما يُعِيقُنا في دراستنا هذه.

فغاية ما أردنا الوصول إليه في هذه الدراسة هو استحداث خوارزمية لتوصيف ما كان على هذا النمط القياسي في قواعد الصَّرف العربيِّ، أمَّا تَرْكُنَا للخماسي وغيره مِمَّا كان جاريا على القياس فالغاية منه اختصار الدراسة وعدم توسيع نطاقها، حتَّى يتسَنَّى للباحثين سهولة النظر فيها، وكذلك تفاديا لعدد احتمالات الأوضاع الداخلة للمعالجة والتي سوف نتطرق إليها في موضعها من هذه الدراسة.

3. التَّوصيف العام لنص التَّصغير ومدخل إلى المستوى البرمجي الحاسوبي:

قبل عملية التوصيف يجب أن نحدد المخطط العام الذي سوف تسير عليه عملية التَّصغير، ويجب أن نراعي في ذلك ما تحتمله اللغة البرمجية التي قمنا بتوظيفها في هذا السياق وهي لغة (جافاسكريبت) (JavaScript)، والأسباب المنطقية التي حَمَلَتْنا على اختيار هذه اللغة دون سواها من اللغات.

فأمَّا ما تحتمله لغة (جافاسكريبت) بالنسبة للغة العربية فإنَّها تدعم جميع الحروف العربية (chars)، وتستطيع قراءتها قراءة واضحة دون أي قارئ مساعد لذلك أو استدعاء مكتبة خاصة، ولكن يجب أن نفهم أنَّ هذه اللغة لا تقرأ المتغير النَّصي العربيّ كما نقرأه نحن في اللغة العربية، فمثلا إذا أنشأنا متغيرا نصيًّا (string variable) قيمته «فعل» بدون تشكيل فإنَّه سوف يَتَرَكَّبُ من ثلاثة حروف (chars)، أمَّا إذا أنشأنا متغيرا نصيًّا آخر قيمته «فَعَلَ» بالتَّشكيل فإنَّهُ سوف يتركَّبُ من ستَّة حروف على هذا التَّرتيب: فاء ثُمَّ فتحة ثُمَّ عَيْن ثُمَّ فتحة ثُمَّ لام ثُمَّ فتحة، ولهذا وجب علينا قبل كل شيء فهم هذه اللغة فهما جيدا حتى نستطيع التعامل معها بما يوافق بُنْيَتها البرمجية.

شكل يوضح آلية عمل المتغير النصي الحامل لقيمة بالحروف العربية في لغة جافاسكريبت

كما يجب كذلك، أن نحصي كل الأوضاع الموجودة في الحروف العربية ونعرف قياسها في لغة (جافاسكريبت)، حتى يتسنى لنا وضع الخوارزميات المناسبة لمعرفة الكلمة المدخَلة للبرنامج هل هي ثنائية أم ثلاثية أم رباعية أم غير ذلك، والشكل الآتي يوضح إحصاء أوضاع الحروف العربية ورموزها جميعا:

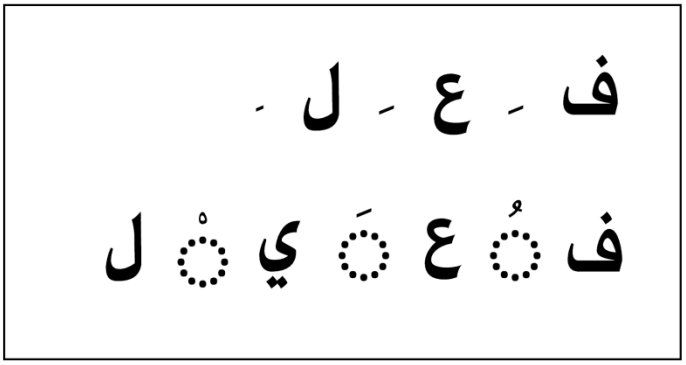
شكل يوضح كل الأوضاع العربية التي طولها حرف واحد في لغة جافاسكريبت

وأمَّا الأسبابُ المنطقيَّة التي حَمَلَتْنا على اختيار هذه اللغة فهي على سبيل الإجمال كالآتي:

- لغة (جافاسكريبت) لغة برمجة نصية مصمَّمة للتفاعل الديناميكي، وهي تعمل جنبًا إلى جنب مع لغة (html) ولغة (css) لتقديم محتوى مميز وتجربة مستخدم عالية.
- تُوَظَّفُ لغة (جافاسكريبت) من طرف المستخدم: (client side)، أي أنَّ أداءها قويٌّ جدًّا وسريع، كونها تُعَالَجُ من طرف المتصفح ولا تخضع لعمليات معالجة من طرف الخادم (Server).
- تتميز بواجهات غنية جدا أي أنَّ المخارج فيها مناسبة جدا لأي تطبيق.

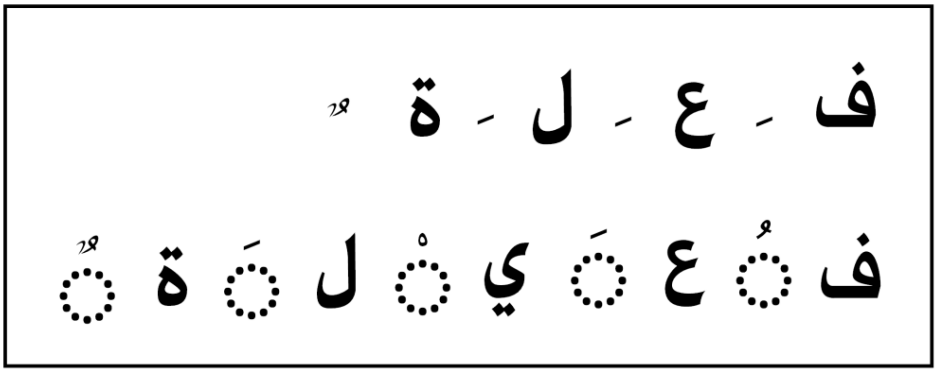
- الإصدارات الحديثة للغة قوية جدا مثل الإصدار الأخير المسمى (ECMAScript6) والذي يحتوي على دوال نصية وتكرارية وميزات وظيفية عالية الكفاءة.
- قدرة اللغة العالية على التعامل مع العناصر من اليمين إلى اليسار وهذا ما يخدم اللغة العربية بالضبط.

وبناء على ما تقدَّم يُمكننا توصيف النَّص الصَّرفي للتصغير، وذلك بتحليل القاعدة تحليلا رياضيا منطقيا ثم النظر في الطريقة المتاحة لذلك؛ فالاسم الثلاثي يُصَغَّر على صيغة (فُعَيْل)، وذلك بأنْ نَضُمَّ الحرف الأوَّل، ونفتح الحرف الثاني، ثم نزيد بعده ياء ساكنة هي التي تسمى ياء التَّصغير، ثم يأتي الحرف الثالث دون تغيير، فنقول: رَجُل ورُجَيْل، أي أنَّ القاعدة بَيِّنة لا لبس فيها ويوضحها الشكل الآتي:



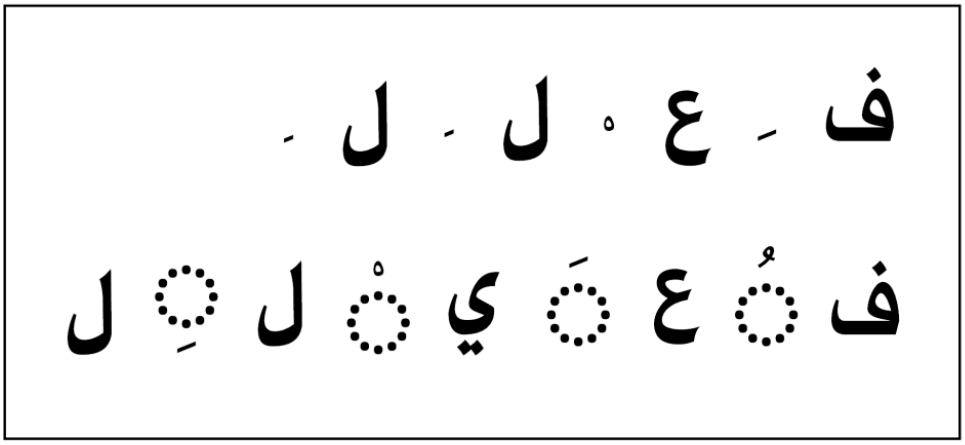
شكل يوضح قاعدة تصغير الثلاثي

فإنْ كان بعد الاسم الثلاثي تاء تأنيث فإنَّها لا تؤثر على هذه العملية، فنقول: بَقَرَة وبُقَيْرَةٌ، ويوضح هذه القاعدة الشكل الآتي:



شكل يوضح قاعدة تصغير الثلاثي المختوم بتاء التأنيث

أمَّا الاسم الرُّباعي:يصغر على صيغة (فُعَيْعِل) أي بأن نَضُمَّ الحرف الأول، ونفتح الحرف الثاني، ثُمَّ نزيد ياء التَّصغير السَّاكنة، ثُمَّ نَكسر الحرف الذي بعدها، فنقول: جَعْفَر وجُعَيْفِر، ويوضح هذه القاعدة الشكل الآتي:



شكل يوضح قاعدة تصغير الاسم الرباعي

4. التوظيفُ البَرمجيّ لتوصيف التَّصغير بلغة جافاسكريبت:

التوظيف البرمجي لعملية التوصيف السابقة سوف يكون عبارة عن برنامج مبني بلغة (جافاسكريبت) (JavaScript)، وقبل الخوض في بناء الخوارزميات المناسبة لهذا الأمر يجب أن نبني الواجهة الأولى التي يتعامل معها المستخدم، وذلك بلغة (html) ولغة (css)، وهي واجهة تفاعلية بالتصميم الذي توضحه هذه الأشكال:

شكل يوضح نتيجة تصميم واجهة برنامج التَّصغير

بعد إعداد الواجهة المناسبة للبرنامج نشرع في تبيان الخوارزمية العامة التي ينطلق منها، وهي أنَّ الكلمة عندما تدخل إلى البرنامج عبر شريط الإدخال تَمُرُّ بثلاث مراحل نبينها كالآتي:

المرحلة الأولى:

في هذه المرحلة تقوم الدَّالة المسمَّاة [scanArabicLetters()] بالتَّحقق من كلِّ حروف القيمة (الكلمة أو غيرها) المدخلة إن كانت حروفا عربية أو غير ذلك كالحروف الأجنبية أو الرموز، وتتم هذه العملية عن طريق مقارنة كلِّ حرف من حروف هذه القيمة المدخلة بحزمة من الحروف العربية وعلامات التشكيل المخزنة في متغير، ثم تقوم هذه الدالة بعد عدة خطوات من معالجة هذه القيمة بإصدار قيمة بولينية (boolean value) في نهاية كل هذه الخطوات كالآتي:

- قيمة صح أو (true): تفيد أنَّ القيمة المدخلة قيمة عربية مطلقة خالية من الرموز أو الحروف الأجنبية.
- قيمة خطأ أو (false): تفيد أنَّ القيمة المدخلة قيمة مركبة من الرموز أو الحروف الأجنبية.

```
function scanArabicLetters(word) {
const arrayLettersArab = ["د", "ج", "ح", "خ", "ه", "ع", "غ", "ف", "ق"
, "ث", "ص", "ض", "ذ", "ش", "س", "ي", "ب", "ل", "ت", "ن", "م", "ك", "ط"
, "ظ", "ز", "و", "ة", "ى", "ا", "آ", "ر", "ؤ", "ء", "ّ", "ٌ", "ٍ", "َ", "ُ", "ِ"
, "ئ", "أ", "إ"];
let resultScanWord = true;
let wordToArray = Array.from(word);
console.log(arrayLettersArab.length);
wordToArray.forEach((letter, index) => {
// console.log(`${index} => ${letter}`);
let letterCompare = arrayLettersArab.indexOf(letter);
// console.log(`${index} => ${letter} => ${letterCompare}`);
while (letterCompare === -1) {
// console.log('bad word');
    resultScanWord = false;
    break;
}
});
return resultScanWord;
}
```

الكود البرمجي للدالة scanArabicLetters()

المرحلة الثانية:

في هذه المرحلة يتم حفظ القيمة البولينية (boolean value) الصَّادرة عن الدالة [scanArabicLetters()] في متغير جديد، ثمَّ يَتِمُّ بناء شرط برمجي على النَّحو التالي:

إذا كانت (القيمة المدخلة تحتوي على قيمة خالية وفارغة) فإنَّه يجب إخبار المستخدم أنَّ حقل الإدخال فارغ لأنَّ القيمة الَّتي حاول تصغيرها قيمة فارغة وخالية (null).

صورة من البرنامج في حالة الحقل الفارغ

أمَّا إذا كانت (القيمة المدخلة قد مرَّت على الدَّالة [scanArabicLetters()] وأصدرت قيمة بولينية صح true) فإنَّ هذا يفيد أنَّ هذه القيمة عبارة عن كلمة عربية وأنَّها قابلة الآن للخضوع إلى الشرط الجديد الذي يُمَكِّنُنَا من تصغيرها.

```
else if (returnresult === true) {
       if ( myInput.value.length === 1 ) {
     } else if ( myInput.value.length >= 2 && myInput.value.length <= 4 ) {
       } else if ( myInput.value.length >= 3 && myInput.value.length <= 6 ) {
      } else if ( myInput.value.length >= 4 && myInput.value.length <= 8 ) {
       } else {         }
```

الكود البرمجي لتحقيق جزء من الشرط

أمَّا إذا كان غير ذلك مِمَّا تَقَدَّم (أي أنَّ القيمة المدخلة قد مَرَّت على الدالة [scanArabicLetters()] وأصدرت قيمة بولينية خطأ false) فإنَّ هذا يفيد أنَّها عبارة عن كلمة غير عربية، ويجب إخبار المستخدم بأنَّ هذه القيمة عبارة عن رموز أو حروف أجنبية لا شأن لها باللغة العربية.

```
else {
    let parentEl = document.querySelector('.section-output');
    let elP = document.createElement('p');
    elP.className = 'tanbih';
    let str ="تنبيه: هناك خطأ في كتابة الكلمة، لا تقبل الرموز ولا تقبل الحروف الأجنبية عن اللغة العربية ";
    let textNode = document.createTextNode(str);
    elP.appendChild(textNode);
    parentEl.appendChild(elP);
    myInput.value = '';
    this.setAttribute('disabled','');
    myInput.onfocus = function() {
    resultButton.removeAttribute('disabled');
    elP.remove();
    this.placeholder = '';
    };
    }
```

الكود البرمجي لتحقيق جزء من الشرط

صورة من البرنامج في حالة إدخال قيمة تحتوي حروف أجنبية أو رموز

المرحلة الثالثة:

لقد مرَّ علينا أنَّ القيمة المدخلة حين تكون كلمة عربية فإنَّها سوف تخضع لشرط جديد مضمن في الشرط الأول، غير أنَّ هذا الشرط يتغير باعتبار طول الكلمة المدخلة (length)، أي أنَّنا نقوم بقياس طول الكلمة وعلى هذا الطول المحدد تتفرع الأفعال المشروطة، ويمكننا توضيح هذا الشرط على النَّحو التالي:

إذا كان (طول الكلمة يساوي 1): نقوم بإخبار المستخدم أنَّ القيمة المدخلة تتركب من حرف واحد، وعليه، إذن، أن يدخل قيمة معتبرة للتصغير.

صورة من البرنامج في حالة إدخال قيمة تحتوي حرفا واحدا

أمَّا إذا كان (طول الكلمة يساوي من 2 إلى 4):فسوف نقوم بإخبار المستخدم أنَّ القيمة المدخلة تتركب من حرفين، أي أنَّ الاسم ثنائي الوضع وثنائي الوضع لا يدعم تصغيره هذا البرنامج كما تقدم في أوَّل البحث.

صورة من البرنامج في حالة إدخال قيمة تحتوي حرفين

وممَّا لا ينبغي إغفاله في هذه الحالة أنَّ القيمة المدخلة التي طولها من 2 إلى 4 تحتمل أوضاعا نستطيع إحصاءها رياضيا، وذلك أنَّ الطول 2 يحتمل 4 أوضاع، والطول 3 يحتمل 8 أوضاع، والطول 4 يحتمل 16 وضعا، وهذه الأوضاع المحتملة كلها في هذا الشرط أي من 2 إلى 4، ولمعرفة طريقة هذا الإحصاء الرياضي الاحتمالي نتطرق إلى هذه المبرهنة الرياضية[4]:

لتكن $A = \{a_1, a_2, \ldots\ldots, a_n\}$ مجموعة ما ذات n عنصرا.

نسمي العينة «ترتيبة بتكرار من الرتبة r من A» (يمكن A أن يكون أكبر من n)، إذا كان يسمح فيها بالتكرار مع مراعاة الترتيب فيها.

مبرهنة:

عدد الترتيبات بتكرار من الرتبة r من n، $T_n^r = n^r$

نطبق هذه المبرهنة لتحديد عدد الحالات الممكنة لظهور الحرف أو الحركة، مع مراعاة الترتيب والتكرار، فقد يتكرر الحرف وكذلك الحركة.

ومثالا على هذا:

لتكن $A = \{s_1, h_1\}$ مجموعة ذات عنصرين نرمز للحرف بـ shakl ونرمز للحركة بـ harfe.

عدد الحالات الممكنة لظهور الحرف والحركة من الرتبة $r = 2$ حسب المبرهنة السَّابقة هو $T_2^2 = 2^2 = 4$.

عدد الحالات الممكنة لظهور الحرف والحركة من الرتبة $r = 3$ حسب المبرهنة السَّابقة هو $T_2^3 = 2^3 = 8$.

عدد الحالات الممكنة لظهور الحرف والحركة من الرتبة $r = 4$ حسب المبرهنة السَّابقة هو $T_2^4 = 2^4 = 16$.

عدد الحالات الممكنة لظهور الحرف والحركة من الرتبة $r = 5$ حسب المبرهنة السَّابقة هو $T_2^5 = 2^5 = 32$.

عدد الحالات الممكنة لظهور الحرف والحركة من الرتبة $r = 6$ حسب المبرهنة السَّابقة هو $T_2^6 = 2^6 = 64$.

عدد الحالات الممكنة لظهور الحرف والحركة من الرتبة $r = 7$ حسب المبرهنة السَّابقة هو $T_2^7 = 2^7 = 128$.

عدد الحالات الممكنة لظهور الحرف والحركة من الرتبة $r = 8$ حسب المبرهنة السَّابقة هو $T_2^8 = 2^8 = 256$.

وتوقفنا عند الرتبة 8 لأنَّ أقصى حدٍّ تحتمله دراستنا هو الاسم الرباعي الذي يتركب من 4 إلى 8 حروف أو علامات تشكيل على حدٍّ أقصى.

نعود مرَّة أخرى إلى الإحصاء الفعلي الذي ذكرناه سابقا من الطول 2 إلى 4، ويفيدنا هذا الإحصاء في انتقاء وفلترة كلِّ الأوضاع التي يحتمل أن يدخلها المستخدم بالنسبة لثنائي الوضع، والتي يقبل المنطق اللغوي بعضها ويرفض البعض الآخر، ولهذا فإحصاء كل الأوضاع مهِمٌّ جدًّا في طريقة بناء الدالة [scanTwoletters ()] التي سوف تقوم بانتقاء كلِّ ما هو مقبول، والتي تقوم أيضا بتمرير الأوضاع الخاصة بالثلاثي والرباعي إلى الدوال الخاصة بها.

```
function scanTwoletters(twoletters) {
    const tachkeel = ["ً","ٌ","ٍ","َ","ُ","ِ","ّ","ْ"];
    let newArrayword = [];
    arrayTwoletters = Array.from(twoletters);
    arrayTwoletters.forEach((letter, index) => {
        if (tachkeel.indexOf(letter) !== -1) {
            newArrayword.push("shakl");
        } else {
            newArrayword.push("harfe");
        }
```

أمَّا إذا كان (طول الكلمة يساوي من 3 إلى 6): فسَوفَ تَدْخُلُ الكلمة إلى الدَّالة [scanThreeletters()]، وقد يُحتمل أنَّ القيمة المدخلة إليها تكون مُمَرَّرَةً من الدَّالة السَّابقة [scanTwoletters()]، لأنَّ هناك وضعا واحدا يدخل في الدالة السابقة الذكر، وهو مصنف في الثلاثي وهو الوضع حرف وحرف وحرف.

عمل هذه الدالة واضح جدًّا فهي إمَّا أن ترفض القيمة المدخلة لكونها قيمة خاطئة، وإمَّا أن تستدعي الدالة [tasgheerAltholathi()] التي تقوم بعملية التصغير لأن القيمة المدخلة تتوفر على جميع شروط الاسم الثلاثي. وتكون النتيجة عندها كما توضحها الصورة الآتية:

صورة من البرنامج توضح عملية تصغير الاسم الثلاثي

```
function scanThreeletters(threeletters) {
    const tachkeel = ["ً","ٌ","ٍ","َ","ُ","ِ","ّ","ْ","ٰ"];
    let newArrayword = [];
    // console.log(tachkeel);
    arrayThreeletters = Array.from(threeletters);
    arrayThreeletters.forEach((letter, index) => {
        // console.log(`letter: ${letter}, index:${index}`);
        if (tachkeel.indexOf(letter) !== -1) {
            newArrayword.push("shakl");
        } else {
            newArrayword.push("harfe");
        }
    });
if ( newArrayword.toString() === ["harfe", "harfe", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "shakl", "harfe", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "harfe", "shakl"].toString() ||
newArrayword.toString() === ["harfe", "shakl", "harfe", "shakl", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "shakl", "harfe", "harfe", "shakl"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe", "shakl"].toString() ||
newArrayword.toString() === ["harfe", "shakl", "harfe",
 "shakl", "harfe", "shakl"].toString() ) {
        // here function of Tasgheer al-tholathi:
        let ismosaghar = tasgheerAltholathi(threeletters);
        console.log('al-tholathi');
        let parentEl = document.querySelector('.section-output');
        let textNode = document.createTextNode(ismosaghar);
        showResult.appendChild(textNode);
        parentEl.appendChild(showResult);
        myInput.value = '';
        resultButton.setAttribute('disabled','');
        myInput.onfocus = function() {
            resultButton.removeAttribute('disabled');
            textNode.remove();
            showResult.remove();
            this.placeholder = '';
        }
} else if ( newArrayword.toString() === ["harfe", "shakl", "harfe", "harfe", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "harfe", "shakl", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "harfe", "harfe", "shakl"].toString() ||
newArrayword.toString() === ["harfe", "shakl", "harfe", "shakl", "harfe", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "shakl", "harfe", "harfe", "shakl", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe", "shakl", "harfe"].toString() ||
newArrayword.toString() === ["harfe", "shakl", "harfe", "harfe", "harfe", "shakl"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe", "harfe", "shakl"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "harfe", "shakl", "harfe", "shakl"].toString() ||
newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe", "harfe", "shakl"].toString() ) {
        scanFourletters(threeletters);
    } else {
        let parentEl = document.querySelector('.section-output');
        let elP = document.createElement('p');
        elP.className = 'tanbih';
        let textNode = document.createTextNode("البرنامج هذا يدعمها لا الاسم بنية أنّ أو الكتابة، صيغة في خطأ هناك :تنبيه")
;
        elP.appendChild(textNode);
        parentEl.appendChild(elP);
        myInput.value = '';
        resultButton.setAttribute('disabled','');
        myInput.onfocus = function() {
            resultButton.removeAttribute('disabled');
            elP.remove();
            this.placeholder = '';
        }
    }
}
```

```
function tasgheerAltholathi(altholathi) {
  let newarray = [];
  let arryString = Array.from(altholathi);
  arryString.forEach(el => {
        if ( el === "ً" || el === "ٌ" || el === "ٍ" || el === "ّ" ||
             el === "ُ" || el === "َ" || el === "ِ" || el === "ْ"  ) {

        } else {
          newarray.push(el);
        }
    });
  let tasgheerString = newarray[0] + "ُ" + newarray[1] + "َ" + "ي" + "ْ" +newarray[2];

      return tasgheerString;
}
```

الكود البرمجي للدالة tasgheerAltholathi()

أمَّا في حالة إذا ما كان (طول الكلمة يساوي من 4 إلى 8): فسَوفَ تَدْخُلُ الكلمة إلى الدَّالة [scanFourletters()]، وقد يُحتمل أنَّ القيمة الدَّاخلة إليها تكون مُمَرَّرةً من الدَّالة [scanTwoletters()] لأنَّ هناك وضعا واحدا يدخل في الدَّالة السَّابقة الذكر، أو من الدالة [scanThreeletters()] لأنَّ هناك وضعا واحدا يدخل في الدالة التي ذكرنا، وعمل هذه الدالة واضح جدًّا وهو إمَّا أن ترفض القيمة الداخلة لكونها قيمة خاطئة أو أنَّها تتوفر على جميع شروط الاسم الرباعي وعندها تستدعي الدالة [tasgheerAlrobaai()]، وهي الدالة التي تقوم بعملية تصغير هذا الاسم الرباعي، أو تقوم بتصغير الاسم الثلاثي المختوم بتاء التأنيث، وتكون النتيجة عندها كما توضحها الصورة الآتية:

صورة من البرنامج توضح عملية تصغير الاسم الثلاثي المختوم بتاء التأنيث

صورة من البرنامج توضح عملية تصغير الاسم الرباعي

```
function tasgheerAlrobaai(alrobaai) {
    let newarray = [];
    let arryString = Array.from(alrobaai);
    arryString.forEach(el => {
        if ( el === "ً" || el === "ٌ" || el === "ٍ" || el === "ِ" ||
             el === "ُ" || el === "َ" || el === "ْ" || el === "ّ"  ) {
        } else {
          newarray.push(el);
        }
    });
    // tholathi makhtom ta'a marbota:
    if ( newarray[3] === 'ة' ) {
        let tasgheerStringTaa = newarray[0] +
            "َ" + newarray[1] + "ْ" + "ي" +
            "َ" + newarray[2] + "ْ" + newarray[3];
        Return tasgheerStringTaa;
    } else {
        Let tasgheerString = newarray[0] +
            "َ" + newarray[1] + "ْ" + "ي" +
            "َ" + newarray[2] + "ِ" + newarray[3];
        Return tasgheerString;
    }
}
```

الكود البرمجي للدالة **tasgheerAlrobaai()**

```
function scanFourletters(fourletters) {
    const tachkeel = ["ً","ٌ","ٍ","َ","ُ","ِ","ّ","ْ","ٓ"];
    let newArrayword = [];
    // console.log(tachkeel);
    arrayThreeletters = Array.from(fourletters);
    arrayThreeletters.forEach((letter, index) => {
        // console.log(`letter: ${letter}, index:${index}`);
        if (tachkeel.indexOf(letter) !== -1) {
            newArrayword.push("shakl");
        } else {
            newArrayword.push("harfe");
        }
    });
    If ( newArrayword.toString() === ["harfe", "harfe", "harfe", "harfe"].toString() ||
         newArrayword.toString() === ["harfe", "shakl", "harfe", "harfe", "harfe"].toString() ||
         newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe", "harfe"].toString() ||
         newArrayword.toString() === ["harfe", "harfe", "harfe", "shakl", "harfe"].toString() ||
         newArrayword.toString() === ["harfe", "harfe", "harfe", "harfe", "shakl"].toString() ||
         newArrayword.toString() === ["harfe", "shakl", "harfe", "shakl", "harfe", "harfe"].toString() ||
         newArrayword.toString() === ["harfe", "shakl", "harfe", "harfe", "shakl", "harfe"].toString() ||
         newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe", "shakl", "harfe"].toString() ||
         newArrayword.toString() === ["harfe", "shakl", "harfe", "harfe", "harfe", "shakl"].toString() ||
         newArrayword.toString() === ["harfe", "harfe", "harfe", "shakl", "harfe", "shakl"].toString() ||
         newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe", "harfe", "shakl"].toString() ||
         newArrayword.toString() === ["harfe", "harfe", "shakl", "harfe", "shakl", "harfe", "shakl"].toString()
 ||
         newArrayword.toString() === ["harfe", "shakl", "harfe", "shakl", "harfe", "harfe", "shakl"].toString()
 ||
         newArrayword.toString() === ["harfe", "shakl", "harfe", "harfe", "shakl", "harfe", "shakl"].toString()
 ||
        newArrayword.toString() === ["harfe", "shakl", "harfe", "shakl", "harfe", "shakl", "harfe", "shakl"].to
String() ) {

        // here function of Tasgheer al-Robaai:
        let ismosaghar = tasgheerAlrobaai(fourletters);
        console.log('roba3i');

        // show result
        let parentEl = document.querySelector('.section-output');
        let textNode = document.createTextNode(ismosaghar);
        showResult.appendChild(textNode);
        parentEl.appendChild(showResult);
        myInput.value = '';
        resultButton.setAttribute('disabled','');
        myInput.onfocus = function() {
            resultButton.removeAttribute('disabled');
            textNode.remove();
            showResult.remove();
            this.placeholder = '';
        }

    } else {
        let parentEl = document.querySelector('.section-output');
        let elP = document.createElement('p');
        elP.className = 'tanbih';
        let textNode = document.createTextNode("تنبيه: البرنامج هذا يدعمها لا الاسم بنية أنّ الكتابة أو صيغة في خطأ هناك");
        elP.appendChild(textNode);
        parentEl.appendChild(elP);
        myInput.value = '';
        resultButton.setAttribute('disabled','');

        myInput.onfocus = function() {
            resultButton.removeAttribute('disabled');
            elP.remove();
            this.placeholder = ''; } }
}
```

الكود البرمجي للدالة scanFourletters()

أمَّا إذا كان (طول الكلمة يساوي من 9 إلى 14): فإنَّ هذه القيمة المدخلة سوف تصنف ضمن الشرط المختلف عن كل الشروط الأخرى (else)، أي أنَّ الاسم إما أن يكون خماسيا أو سداسيا أو سباعيا أو مجموعة حروف لا يتعدى طولها 14 حرفا، لأن البرنامج لن يفوق إدخاله 14 حرفا، وعليه في هذه الحالة إخبار المستخدم أنَّ هناك خطأ في صيغة الكتابة أو أنَّ بنية الاسم لا يدعمها هذا البرنامج كما توضحه هذه الصورة:

صورة من البرنامج توضح النتيجة إذا كان عدد حروف القيمة المدخلة من 9 إلى 14 حرف

هذه إذن هي الخوارزمية العامة التي سار عليها برنامج التوصيف، وكما تتبعنا فيما سبق أعلاه فإنَّ نص توصيف قواعد التصغير قد تحوسَبَ كليًّا، كما كان مسَطَّرًا له في بداية البحث بلغة (جافاسكريبت).

5. خاتمة:

من خلال هذا البحث توصلنا إلى النتائج التالية:

يواجه توصيف قواعد التَّصغير الصَّرفية كثيرا من العقبات والتحديات الحاسوبية.

تستطيع قواعد الصرف العربيِّ حمل الكثير من المفاهيم المنطقية الرياضية والمفاهيم البرمجية.

تستطيع لغة (جافاسكريبت) بناء ما يمكن توصيفه في الدرس الصرفيِّ العربيِّ.

تتضمن المفاهيم اللسانية الحاسوبية الكثير من مفاهيم الإحصاء والاحتمالات الرياضيَّة.

6. الهوامش:

1- مصطفى الغلاييني، جامع الدروس العربية، المكتبة العصرية، لبنان، ط28، 1993م، الجزء الثاني، ص84-95.

2- عبده الراجحي، التطبيق الصرفي، دار النهضة العربية، لبنان، الطبعة الأولى،1999م، ص131.

3- المرجع نفسه، ص132 -133.

4- سيمور ليبشتز ومارك ليبسون، الرياضيات المتقطعة، ترجمة انتصارات محمد حسن الشبكي، الدار الدولية للاستثمارات الثقافية، جمهورية مصر العربية، الطبعة الأولى، 2006م، ص63.

7. قائمة المراجع:

1. مصطفى الغلاييني، جامع الدروس العربية، المكتبة العصرية، لبنان، الطبعة الثامنة والعشرون، 1993م.
2. التطبيق الصرفي، عبده الراجحي، دار النهضة العربية، لبنان، الطبعة الأولى،1999م.
3. سيمور ليبشتز ومارك ليبسون، الرياضيات المتقطعة، ترجمة انتصارات محمد حسن الشبكي، الدار الدولية للاستثمارات الثقافية، مصر، الطبعة الأولى، 2006م.