
**

*

2008/1/23:

2008/5/8:

B+

B+

Abstract

Usually, the natural language grammar is represented as a rules inside the program of natural language processing (NLP), but with a language such as the Arabic language that have a flexible syntax, using this representation will produce a large number of these rules that lead to increase the search time required and to increase the size of the program, so in this paper we proposed a new representation method. We represent the Arabic language grammar as logical terms in the knowledge base, and we use the B+tree to store these terms since it is suitable with large data, so we reach the required syntactical term in a little time by using the suitable searching method that b+tree provide it.

In this paper we choose the Augmented Transition Network(ATN) to represent the language grammar since it is the most suitable parsing technique for a language such as the Arabic language.

[1]

*

Context Free)

(Grammar

.1

:[3]

الجملة ← فعل + فاعل + مفعول به

فعل ← كتب

فاعل ← اسم

مفعول به ← اسم

اسم ← أداة تعريف + اسم

اسم ← ولد

اسم ← درس

.2

.3

(Natural Language Processing)

(States)

.(Transitions)

()

.()

" "

" "

(Phonology)

(Syntax)

(Morphology)

(Semantics)

.[2](Lexicon)

(Pragmatics)

- RTN)

(Recursive Transition Network

-ATN)

(Augmented Transition Network

1

(Grammar)

ATN

.[4]1970 William A. Woods

¹ القواعد (Grammar) للغة هي المفاهيم التي تحدد الجمل المسموحة في تلك اللغة.

(Registers)

)
 (...) :ATN
 Saad builds houses in Iraq.
 : ATN
 Path [[NP Saad][VP builds [NP houses][PP in [NP Iraq]]]]
 :Int_list (Saad)
 (builds houses in Iraq)
 .(... :Roles (Houses) (builds)
 : (in Iraq)
 .(noun phrase) :NP
 .(verb phrase) :VP
 preposition) :PP
 .(phrase

B+ (Terms)
(Logical Terms)

(B+ tree)
 B list String
 nested list
 [2] B+
 B+
 (data structure) .(Compound term
 (Predicate Calculus)

B+
 index to)
 (database .[5]
 .(indices) "bnf"
 :
 :New States
 B+)
 (leaves) (key) .(ATN
 :Path
 (reference number)

"	:	2-8	(Count)
"	:	3-8	
.(19)	:		(Path)
:	:	.9	
bnf(NewStateList,List,IntList,R	:		(ATN)
oles)	:	.10	
	:		.(StateList)
	:		StateList .1
:	:	.11	:
	:	1-11	.(2) : 1-1
	:		(s) : 2-1
	:	.(12)	.StateList
"	:	2-11	.2
"	:		.State StateList
.(19)	:	3-11	.3
	:	.12	:)
.Roles	:		(
	:	.13	.(4) : 1-3
.IntList	:		LastType : 2-3
NextState	:	.14	.
.NewStateList	:		.(5) 3-3
NextState	:	.15	Path .4
.StateList	:		:
	:	.16	LastType : 1-4
.List SubType	:		.Path
Path	:	.17	
.Count SubType	:		.(5) 2-4
	:	.18	.(7) : 3-4
	:		.5
	:		LastType Type
	:	.19	.State
	:		.6
. ATN	:		:
	:		.(7) : 1-6
	:		.(9) : 2-6
(Rules)	:		.7
	:		.State Type
	:		.8
)	:		:
	:	.(9)	: 1-8

.4

ATN

(

.(Rules)

(Rules)

B+

B+ tree

(1)

(2)

.1

B+

ATN

.()

.2

B+

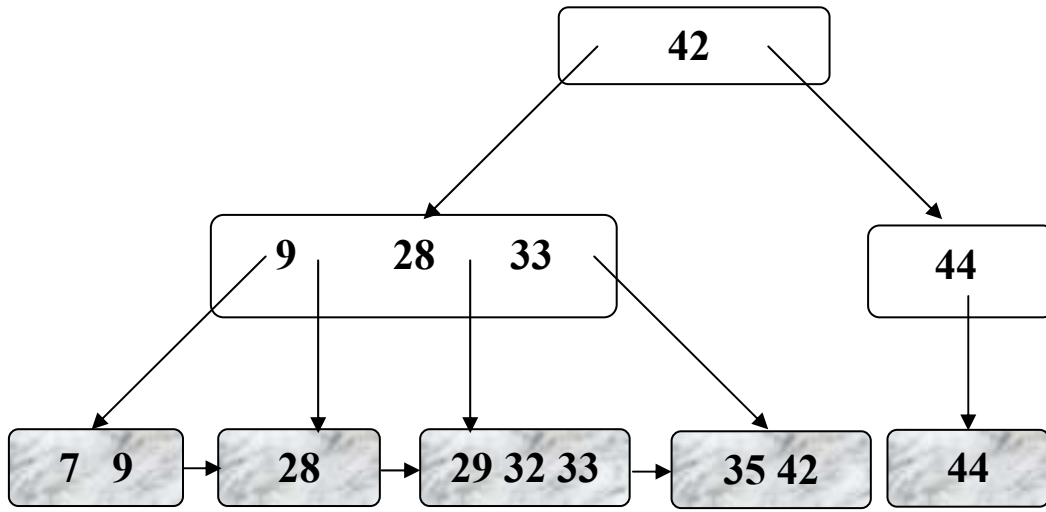
.3

مثال	المفتاح	الأدوار الوظيفية المتوقعة للحالة الحالية	قائمة Path	رموز الحالة الحالية المتوقعة	الحالات المتوقعة التالية	نوع الكلمة أو المركب الحالي	الحالة الحالية	بيانات قواعد اللغة العربية
يذهب	s,[1,0]	فعل	[]	[1]	s_1	فعل لازم	s	bnf([s_1],[1],[1],[1],"فعل")
طالب	s_1,[2,7]	منفذ	[[1]]	[2]	s_2	اسم فاعل	s_1	bnf([s_2],[2],[7],[1],"منفذ")
العلم	s_2,[2,1]	مضاف اليه	[[2],[1]]	[]	s_2	اسم علم	s_2	bnf([s_2],[1],[1],[1],"مضاف اليه")
المجتهد	s_2,[3,1]	صفة	[[2],[1]]	[]	s_2	صفة	s_2	bnf([s_2],[1],[5],[1],"صفة")
النشيط	s_2,[3,1]	صفة	[[2],[1]]	[]	s_2	صفة	s_2	bnf([s_2],[1],[5],[1],"صفة")
إلى	s_2,[6,4]	حرف جر	[[2],[1]]	[6,4]	n_2	حرف جر	s_2	bnf([n_2],[6,4],[3],[1],"حرف جر")
المدرسة	n_2,[2,5],[6,4]	شبه مجرور	[[2],[1],[6,4]]	[]	s_2	اسم مكان	n_2	bnf([s_2],[1],[18,2],[1],"شبه جملة")
في	s_2,[6,4]	حرف جر	[[2],[1]]	[6,4]	n_2	حرف جر	s_2	bnf([n_2],[6,4],[3],[1],"حرف جر").
الصباح	n_2,[2,1],[6,4]	شبه مجرور	[[2],[1],[6,4]]	[]	s_2	اسم علم	n_2	bnf([s_2],[1],[18,2],[1],"شبه جملة").
الباكر	s_2,[3,1]	صفة	[[2],[1]]	[]	s_2	صفة	s_2	bnf([s_2],[1],[5],[1],"صفة").
بالسيارة	s_2,[2,5],[6,4]	شبه مجرور	[[2],[1]]	[]	s_2	حرف جر + اسم آلة	s_2	bnf([s_2],[1],[18,2],[1],"شبه جملة").
جملة مقبولة	s_2,[2],[1]	-	[[2],[1]]	[]	final	فعل، منفذ	s_2	bnf([final],[1],[1],[1]).

جدول (1) يمثل خط سير المعرب مع الجملة العربية (يذهب طالب العلم المجتهد النشيط إلى المدرسة في الصباح الباكر بالسيارة)

))
.(.(
	B+

(2)



(B+ Tree)

(1)

" .1
 " . 2000 .1988 /
 " .4 " .2
 " -
 " . 2000 -
 " .5 . 2003
 " .3
 " -
 " . 2001