



الجامعة الافتراضية السورية
SYRIAN VIRTUAL UNIVERSITY

الذكاء الصناعي

الدكتورة أميمة دكاك



Books

الذكاء الصناعي

الدكتورة أميمة دكّاك

من منشورات الجامعة الافتراضية السورية

الجمهورية العربية السورية 2018

هذا الكتاب منشور تحت رخصة المشاع المبدع – النسب للمؤلف – حظر الاشتقاق (CC– BY– ND 4.0)

<https://creativecommons.org/licenses/by-nd/4.0/legalcode.ar>

يحق للمستخدم بموجب هذه الرخصة نسخ هذا الكتاب ومشاركته وإعادة نشره أو توزيعه بأية صيغة وبأية وسيلة للنشر ولأية غاية تجارية أو غير تجارية، وذلك شريطة عدم التعديل على الكتاب وعدم الاشتقاق منه وعلى أن ينسب للمؤلف الأصلي على الشكل الآتي حصراً:

أميمة دكّاك، الذكاء الصناعي، من منشورات الجامعة الافتراضية السورية، الجمهورية العربية السورية، 2018

متوفر للتحميل من موسوعة الجامعة <https://pedia.svuonline.org/>

Artificial Intelligence

Oumaima Dkkak

Publications of the Syrian Virtual University (SVU)

Syrian Arab Republic, 2018

Published under the license:

Creative Commons Attributions- NoDerivatives 4.0

International (CC-BY-ND 4.0)

<https://creativecommons.org/licenses/by-nd/4.0/legalcode>

Available for download at: <https://pedia.svuonline.org/>



الفهرس

- 1 الفصل الأول: مدخل إلى الذكاء الصناعي
- 1- الذكاء 1
- 2- الذكاء الصناعي 1
- 3- مجالات الذكاء الصناعي 2
- 4- ما هو الحاسوب الذكي 3
- 5- مقارنات عامة 4
- 1- ما الفرق بين الذكاء الطبيعي والذكاء الصناعي؟ 4
- 2- ما الفرق بين المعالجة الحاسوبية والمعالجة البشرية للمعطيات؟ 4
- 3- ما الفرق بين الذكاء الصناعي والبرمجة التقليدية؟ 5
- 6- هندسة المعرفة 5
- 1- نطاق المعرفة 6
- 2- ما هي هندسة المعرفة؟ 6
- 3- طرق هندسة المعرفة 6
- 7- فلسفة تمثيل المعرفة 7
- 1- الدور الأول: إيجاد البديل عن المعرفة بهدف المحاكمة عليها. 8
- 2- الدور الثاني: إبداع مفاهيم عامة في المعرفة Ontologies 8
- 3- الدور الثالث: اعتماد نظرية تجزيئية للمحاكمة الذكية. 9
- 4- الدور الرابع: إيجاد وسط ليعبر الناس عن معرفتهم. 11
- 5- الدور الخامس: إيجاد وسيط ليعبر الناس عن معرفتهم 11
- 8- مراجع البحث 11
- 12 الفصل الثاني: حل المشاكل باستخدام الفرضيات وحساب الإسناديات
- 1- استخدام المنطق في المحاكمة 12
- 2- حساب الفرضيات 13
- 1- اللغة 13
- 2- قواعد الاستدلال 14
- 3- الدلالة 15
- 4- الحل 18
- 5- الحل بالنقض 19
- 6- عبارات هورن 20
- 3- حساب الإسناديات Predicates 21
- 1- اللغة 21
- 2- قواعد الاستدلال 22

- 22.....3- التوحيد
- 25.....4- الحل
- 26.....5- استخراج الجواب
- 27.....4- تمارين
- 28.....5- مراجع البحث

الفصل الثالث: طرق أخرى لتمثيل المعرفة والنظم الخبيرة

- 29.....1- نظم قواعد الإنتاج
- 29.....1- السلسلة الأمامية
- 30.....2- السلسلة الخلفية
- 30.....3- مقارنة بين السلسلة الأمامية والسلسلة الخلفية
- 31.....4- مشاكل قواعد الإنتاج وحلول
- 32.....2- شبكات الدلالة
- 38.....3- ترابط المفاهيم
- 41.....4- الأطر Frames
- 46.....5- السيناريو
- 49.....6- تمارين
- 50.....7- مراجع البحث

الفصل الرابع: إيجاد الحلول الموضوعية وخوارزميات البحث

- 51.....1- فضاء الحالات
- 55.....2- البحث في فضاء الحالات
- 59.....3- بيان المسائل الجزئية
- 60.....4- الشجرة and-or
- 61.....5- البحث التجريبي
- 63.....6- مراجع البحث

الفصل الخامس: المعرفة والتفكير المتلبسين

- 64.....1- مراجعة الاحتمالات
- 67.....2- قاعدة بايز والمحكمة باستخدامها
- 70.....3- الشك
- 71.....4- عوامل اليقين
- 72.....5- المنطق العائم
- 74.....6- مراجع البحث

75	الفصل السادس: التعلم
75	1- مقدمة
76	2- التعلم الاستقرائي: الشبكات العصبونية
79	3- التعلم الاستنتاجي عبر الأمثلة
84	4- الألعاب
86	5- مراجع البحث
78	الفصل السابع: التواصل والإدراك والفعل
87	1- مقدمة
87	2- حلقة التحسس/ التخطيط/الفعل
88	3- تعلم دوال تجريبية
91	4- الجوائز عوضاً عن الأهداف
92	5- التخطيط في الألعاب ذات اللاعبين
93	6- إجراء min-Max
98	7- مراجع البحث
99	الفصل الثامن: استشراف المستقبل: الخوارزميات المتطورة
99	1- مقدمة
99	2- الخوارزميات الجينية
102	3- البرمجة الجينية
103	4- نظم ذكية هجينة
104	5- مراجع البحث

:

.1

-

:()
:



.2

learning

reasoning

perception

acting

communicating

)

(



.3

-
-
-
-
-
-
-
-

.4

Data



Information



.false true



. "fuzzy logic"

- Switches



Turing

Turing

(B)

(A)

(C) Interrogator

.()

:

."A Y B X" "B Y A X"

Y X

: B A

()

X

:C

C

A

A

A

X

Turing

B A

(Turing)

)

Joseph Weizenbaum

ELISA

(

Mauldin

JULIA

.5

:

المعالجة الصناعية (الحاسوبية)

المعالجة البشرية

تستخدم رموزاً لتمثيل مفاهيم المسألة ومعالجتها بهدف اتخاذ قرارات	تستخدم أرقاماً، ومحارف
ثمة تمثيل مجرد للمعرفة ¹ ، وجداول ² يجري تعلمها لحل المسائل	تستخدم الذاكرة والملفات لتخزين المعلومات والبرامج.
تستخدم برمجة منطقية وغيرها	تستخدم خوارزميات
تستخدم الكسبيات (Heuristics) والاستدلال ومواءمة الأشكال لاتخاذ قرارات أعقد ناجمة عن الخبرة	تنتج قرارات بسيطة...

c) (Jess clips C++ lisp prolog C# Pascal)
 (...)

:

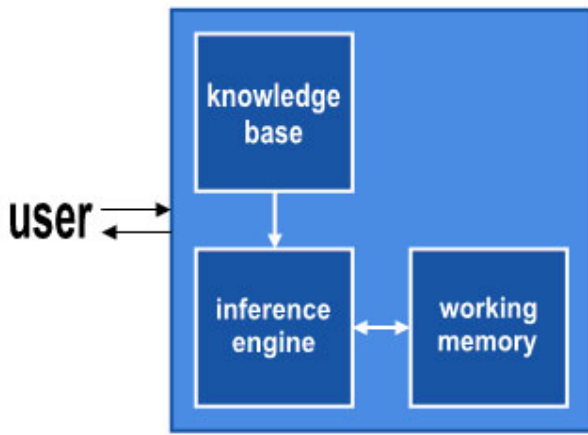
الذكاء الصناعي	البرمجة التقليدية
يقوم بمعالجة رموز	تقوم بمعالجة حسابية
قد لا يكون الدخل والخرج معروفين تماماً في النظم الذكية	الدخل والخرج معرفان تماماً بالخوارزميات
البحث عن الحل تجريبي	ثمة خوارزميات للبحث
التركيز على المعرفة	التركيز على المعطيات والمعلومات
فصل التحكم عن المعطيات، إضافة معطيات جديدة مستقلة عن إضافة أدوات محاكاة وتحكم جديدة	المعطيات مدمجة مع أدوات التحكم، وأي تغيير يتطلب إعادة ترجمة البرامج قبل التنفيذ
سهولة التحديث نسبياً	صعبة التحديث للسبب السابق





:

:



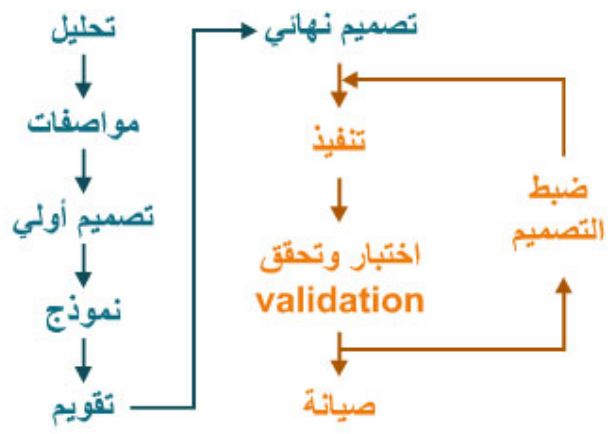
مكونات نظام خبير (نظم قواعد المعرفة)

knowledge base

inference engine

Working Memory

دورة حياة هندسة المعرفة



.7

.1

.() Ontologies	.2
()	.3
	.(
	:	.4
	:	.5
	:	
	:	
	:"	
	:	
	:"	
)	(...)	
	(...)	
	:	
	(..)	
	:	
	...	
	:	
	:)	
	(...)	

(...Lisp, Prolog, CLIPS)

)
MYCIN

Frames

(if, then

INTERNIST

:

:



:



:



.PROLOG

)

:



.(

:



)

.(agents

ANN

:



.(

)

:



Mathematical	Psychology	Biology	Statistics	Economics
Logic				
Aristotle				
Descartes				
Boole	James		Laplace	Bentham, Pareto
Frege		Bernoulli	Friedman	
Peano				
	Hebb	Lashley	Bayes	
Goedel	Bruner	Rosenblatt		
Post	Miller	Ashby	Tversky	Von Neumann
Church	Newell	Lettvin	Kahneman	Simon
Turing	Simon	McCulloch, Pitts		Raiffa
Davis		Heubel, Weisel		
Putnam				
Robinson				
LOGIC	SOAR	CONNECTIONIS M	Causal networks	Rational agents
PROLOG	Knowledge-based systems	A-life		
	Frames			

Views of Intelligent Reasoning and Their Intellectual Origins

:

()



:

Soundness

Abduction

-

.matching

()



frames

(...)

) **.stereotypes, scripts**

!

:

:

...

.Taxonomic

) ()

(

epistemology

:

...

:



.8

- Giarratano & Riley "Expert Systems: Principles and Programming", 3d Edition, 1998
- "What is Knowledge Representation" R. Davis et al. (on the web)

:

.1

) present
 future .(
 "reasoning" " (.)
 "planning" "

"symptoms" " "causes" "
 "expert systems"
 :
 () :

(TRUE,
 : FALSE)

(BAT_OK) x_1
 (LIFTABLE) x_2
 (MOVES) x_3
 ()
) MOVES () BAT_OK
 LIFTABLE .LIFTABLE (

1 LIFTABLE BAT_OK
 0 MOVES 1 MOVES

BAT_OK .0 () LIFTABLE BAT_OK
 .0 LIFTABLE 1

language

inference

propositional calculus

BAT_OK ^ :

LIFTABLE \supset MOVES

." " \supset " " ^

:

.()

:

propositional calculus

first order predicate

.(FOPC) calculus

.2

:

(False True) F T :atoms

.(P, Q, R,..., P1, P2, ON_A_B,... :)

." " " " " " " " $\neg \supset \wedge \vee$:connectives

: (wffs) well-formed formulas

.P3 R P :



:

$\omega_2 \quad \omega_1$



$:\omega_2 \quad \omega_1$

$(\omega_2 \quad \omega_1) \quad \omega_1 \vee \omega_2$

$(\omega_2 \quad \omega_1) \quad \omega_1 \wedge \omega_2$

$(\quad) \quad \omega_1 \supset \omega_2$

$(\omega_1) \quad \neg \omega_1$

:

$(P \wedge Q) \supset \neg P$

$P \supset \neg P$

$P \vee P \supset P$

$(P \supset Q) \supset (\neg Q \supset \neg P)$

$\neg \neg P$

$\omega_1 \supset \omega_2 \quad \omega_1$

.Literal \neg



consequent ω_2

antecedent

$P \supset \neg \neg :$



$(P \wedge Q) \supset \neg R$

$\neg R \quad Q \quad P$

$(P \wedge Q)$

$(P \wedge Q) \supset \neg R \quad R$

rules of

.inference

:

$(\beta \quad \alpha) \quad \alpha \quad \gamma$

:

$(\omega_1 \supset \omega_2 \quad \omega_1) \quad \omega_2$



.(modus ponens

$(\wedge) \quad \omega_2 \quad \omega_1 \quad \omega_1 \wedge \omega_2$



$(\quad \wedge) \quad \omega_1 \wedge \omega_2 \quad \omega_2 \wedge \omega_1$



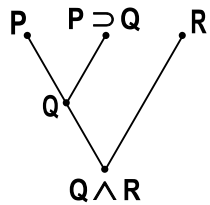
(\wedge) $\omega_1 \wedge \omega_2$ ω_1 ω_2 $\omega_1 \wedge \omega_2$ ω_1
 (\vee) $\omega_1 \vee \omega_2$ ω_1 ω_2 $\omega_1 \vee \omega_2$ ω_1
 (\neg) $\neg(\neg\omega_1)$ ω_1 ω_1 $\neg(\neg\omega_1)$ ω_1

ω_n (deduction) Proof $\{\omega_1, \omega_2, \dots, \omega_n\}$
 Δ ω_1 Δ
 (\quad)
 Δ theorem ω_n Δ ω_n
 $\Delta \models \omega_n$ $:\Delta$ ω_n
 $:$
 $:\Delta$

$\Delta = \{P, R, P \supset Q\}$

$Q \wedge R$
 $\{P, P \supset Q, Q, R, Q \wedge R\}$

Δ wff



semantics

()

propositions

"The Battery is charged" " ¹BAT_OK

.interpretation

.denotation

P α .False True

¹ لسنا مضطرين إلى استخدام سلسلة حرفية سهلة التذكر، إذ يمكن الاستعاضة عنها بسلسلة حرفية أخرى.

.False P True α
 True Bat_Ok :
 F True T
)
 - .False
 (!

(X1)

ω_1	ω_2	$\omega_1 \wedge \omega_2$	$\omega_1 \vee \omega_2$	$\neg \omega_1$	$\omega_1 \supset \omega_2$
True	True	True	True	False	True
True	False	False	True	False	False
False	True	False	True	True	True
False	False	False	False	True	True

True R False Q False P
 :
 [(P \supset Q) \supset R] \supset p
 (P \supset Q) \supset R True P \supset Q
 .False False P True
 model .True

.()
 P \wedge \neg P F ()
 : .(True)
 {P \vee Q, P \vee \neg Q, \neg P \vee Q, \neg P \vee \neg Q}

True ²valid

:

$P \supset P (\neg P \vee P)$
T
 $\neg (P \wedge \neg P)$
 $Q \vee \neg Q$
 $[(P \supset Q) \supset P] \supset P$
 $P \supset (Q \supset P)$

equivalent

:

:DeMorgan

$\neg (\omega_1 \vee \omega_2) \equiv \neg \omega_1 \wedge \neg \omega_2$
 $\neg (\omega_1 \wedge \omega_2) \equiv \neg \omega_1 \vee \neg \omega_2$

:Contrapositive

$(\omega_1 \supset \omega_2) \equiv (\neg \omega_2 \supset \neg \omega_1)$

:

$\omega_2 \omega_1$

$(\omega_1 \supset \omega_2) \wedge (\omega_2 \supset \omega_1)$

$(\omega_1 \supset \omega_2) \wedge (\omega_2 \supset \omega_1) \quad \omega_1 \equiv \omega_2$

True

$\omega \quad \Delta \quad \omega \quad \omega$

Δ

True

Δ

$\Delta \vDash \omega$

$\vDash \Delta$

$\{P\} \vDash P$
 $\{P, P \supset Q\} \vDash Q$
 $F \vDash \omega$

ω

$P \wedge Q \vDash P$

()

Δ

ω

Δ

ω

$$\Delta \quad \omega \quad \Delta \models \omega$$

$$\Delta \quad \omega \quad \Delta \quad \omega$$

$$) \Delta \quad \omega$$

)

(

(

Resolution

$$\begin{array}{l} \{-\lambda\} \cup \Sigma_2 \quad \{\lambda\} \cup \Sigma_1 \quad \lambda \\ \lambda \quad (\quad) \Sigma_2 \quad \Sigma_1 \quad \Sigma_2 \quad \Sigma_1 \\ \Sigma_1 \cup \Sigma_2 \end{array}$$

$$.R \vee Q \quad \neg P \vee Q \quad R \vee P \quad \leftarrow$$

$$.P \supset Q \quad \neg R \supset P$$

$$\neg R \supset Q$$

chaining

$$.R \vee Q$$

$$R \supset P$$

$$:P \quad \neg R \vee P \quad R \quad \leftarrow$$

$$Q \quad :Q \vee R \vee S \vee W \quad P \quad \neg P \vee Q \vee W \quad P \vee Q \vee R \vee S \quad \leftarrow$$

$$.(\quad)$$

$$\lambda \quad (\quad) \neg \lambda \quad (\quad) \lambda \quad \leftarrow$$

$$\neg \lambda \quad \lambda \quad : \quad \neg \lambda \quad \lambda \quad \mathbf{F} \quad \neg \lambda$$

)

.(

$$\neg (P \supset Q) \vee (R \supset P) :$$

$$:\vee \quad .1$$

$$\neg (\neg P \vee Q) \vee (\neg R \vee P)$$

$$(\neg P \vee Q \quad P \supset Q) \neg \quad .2$$

:

$$(P \wedge \neg Q) \vee (\neg R \vee P)$$

.3

$$\neg (P \wedge Q) = (\neg P) \vee (\neg Q)$$

$$\neg (P \vee Q) = (\neg P) \wedge (\neg Q)$$

:

$$(P \vee \neg R \vee P) \wedge (\neg Q \vee \neg R \vee P)$$

:

$$(P \vee \neg R) \wedge (\neg Q \vee \neg R \vee P)$$

$$:(\quad)$$

$$\{ (P \vee \neg R), (\neg Q \vee \neg R \vee P) \}$$

:

Δ

wff

$$.(\quad)$$

Δ



w



. Γ



Γ



:

BAT_OK (1)

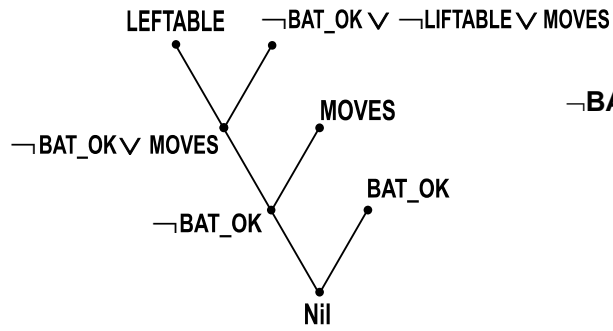
\neg MOVES (2)

$BAT_OK \wedge LIFTABLE \supset MOVES$ (3)

\neg BAT_OK \vee \neg LIFTABLE \vee MOVES (4)

\neg LIFTABLE

LIFTABLE (5)



\neg BAT_OK \vee MOVES (5) (4) (6)

\neg BAT_OK (6) (2) (7)

Nil (7) (1) (8)

:Refinement

()

.Q P Fact

.(Rule)

.P \wedge Q \supset R

wff

Goal

.P^Q ⊃

predicates

.3

B

.C
ON_B_C ON_A_B

ON_B_C

B

P124

()

.Q23

()

ON_B_C ⊃ ¬CLEAR_C

C B) C

CLEAR_C

(C

y x On (x,y) ⊃ ¬Clear (y)

:

.object constants

Aa, 13B, John, :

.EiffelTower

:

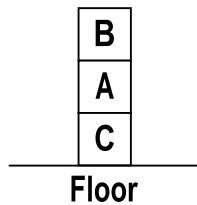
function constants

distanceBetween(Damascus, Homs)
times (5, 4)

relation constants
 Son(Radwan, :
 .Anas)

times (4, x):

() [] ∃ ∀ ⊃ ⊃ ⊃ ⊃ ⊃ ⊃



- Block (A) ●
- Block (B) ●
- Block (C) ●
- Floor (F1) ●
- On (B,A) , On (A,C).. ●
- Clear (B) ●

A	A
B	B
C	C
Floor	F1
On = {<B,A>, <A,C>, <C,Floor>}	On
Clear = {}	Clear

:) (

(∀x)P(x,f(x),B) :Universal Instantiation

.x A x P(A,f(A),B)

(∀x)Q(A,g(A),x) :Existential Generalization

(∃y)(∀x)Q(y,g(y),x)

إذا لم تكن صيغة محققة مهما يكن متحولها فهذا يعني أنه يوجد مثال لا تتحقق من أجله الصيغة. $\neg(\forall\xi) \omega(\xi) \equiv (\exists\xi) \neg\omega(\xi)$
 إذا لم يوجد مثال تتحقق الصيغة من أجله فهذا يعني أنها غير محققة مهما يكن متحولها. $\neg(\exists\xi) \omega(\xi) \equiv (\forall\xi) \neg\omega(\xi)$
 يمكن تغيير المتحول الأصم للمكمي الشمولي \forall دون أن تتأثر دلالة الصيغة. $(\forall\xi) \omega(\xi) \equiv (\forall\eta) \omega(\eta)$

()

$$\neg\lambda(\tau) \quad \left(\begin{array}{c} \xi \\ \tau \end{array} \right) \quad \lambda(\xi) \quad \left(\begin{array}{c} \xi \\ \tau \end{array} \right)$$

Resolvent

$$f(y) \quad x \quad \neg P(x,A) \vee R(x,C) \vee S(A,B) \quad P(f(y),A) \vee Q(B,C)$$

$$\neg P(f(y),A) \vee R(f(y),C) \vee S(A,B)$$

R(f

$$P(f(y),A)$$

$$\neg R(f(y),C) \vee S(A,B) \vee Q(B,C)$$

unification

$$P[x,f(y),B]$$

substitution instance

$$P[z,f(w),B]$$

$$P[x,f(A),B]$$

$$P[g(z),f(A),B]$$

$$P[C,f(A),B]$$

alphabetic variant

$$P[x,f(y),B]$$

ground instance

$$\tau_i/\xi_i \quad \mathbf{s} = \{\tau_1/\xi_1, \tau_2/\xi_2, \dots, \tau_n/\xi_n\}$$

τ_i

ξ_i

P[x,f(y),B]				التعبير
P[z,f(w),B]	P[x,f(A),B]	P[g(z),f(A),B]	P[C,f(A),B]	منتسخات التعويض
s1 = {z/x, w/y}	s2 = {A/y}	s3 = {g(z)/x, A/y}	s4 = {C/x, A/y}	التعويضات المستخدمة

s1

s2

s1s2

s2 s1

s2 s1

s1

s2

$$\omega(s1)s2 = \omega(s1s2)$$

ω

s1s2

ω

$$\omega(s1s2)s3 = \omega(s1(s2s3))$$

$$: \quad s2 = \{A/y\} \quad s1 = \{f(y)/x\}$$

P(x,y)

ω

$$\omega(s_1s_2)=[P(x,y)]\{f(A)/x,A/y\}=P(f(A),A) \quad (\omega s_1)s_2=[P(f(y),y)]\{A/y\}=P(f(A),A)$$

$$.s_1s_2=s_2s_1 \quad :$$

s2 s1 ω

$$\omega(s_1s_2)=P(f(A),A)$$

$$\omega(s_2s_1)=[P(x,y)]\{A/y,f(y)/x\}=P(f(y),A)$$

{ω_i} s
 s unifiable {ω_i} .{ω_i}s
 {ω_i} unifier s ω₁s = ω₂s = ω₃s = ...
 s={A/x,B/y} .singleton
 s={A/x,B/y} .{P[A,f(B),B]} {P[x,f(y),B],P[x,f(B),B]}
 {P[x,f(y),B],P[x,f(B),B]}

most () {ω_i} g A x ()
 s' {ω_i}s {ω_i} s general unifier (mgu)
) .{ω_i}s={ω_i}gs'
 .(

.()

UNIFY

list-structured
 -P(x,f(A,y)) -P (-Px(f A y))
 (f A x)

.disagreement set UNIFY
) W
 W W (

$$\{(-Px(f A y)) , (-Px(f z B))\} .W$$

.{A,z}

.A/z

UNIFY

.{P[A,f(B),B]}

{P[A,f(y),B],P[x,f(B),B]}:

s={A/x,B/y}

:

UNIFY (Γ)

- 1 $k \leftarrow 0; \Gamma_k \leftarrow \Gamma; \sigma_k \leftarrow \varepsilon$
- 2 if Γ_k is singleton, exit with σ_k , the mgu of Γ , otherwise continue
- 3 $D_k \leftarrow$ the disagreement set of Γ_k
- 4 If there exists elements v_k, t_k in D_k such that v_k is a variable that does not occur in t_k continue. Otherwise exit with failure; Γ is not unifiable
- 5 $\sigma_{k+1} \leftarrow \sigma_k\{t_k/v_k\}; \Gamma_{k+1} \leftarrow \Gamma_k\{t_k/v_k\}$
- 6 $k \leftarrow k+1$
- 7 Go to 2

حيث Γ مجموعة قوائم مهيكلة من التعابير

الخطوة الابتدائية. (ε هو التعويض الفارغ)

إذا كانت Γ_k أحادية، أخرج من الإجراء مرجعاً σ_k وهي الموحد الأعم لـ Γ ، وإلا فتابع

نضع في D_k مجموعة عدم التوافق لـ Γ_k

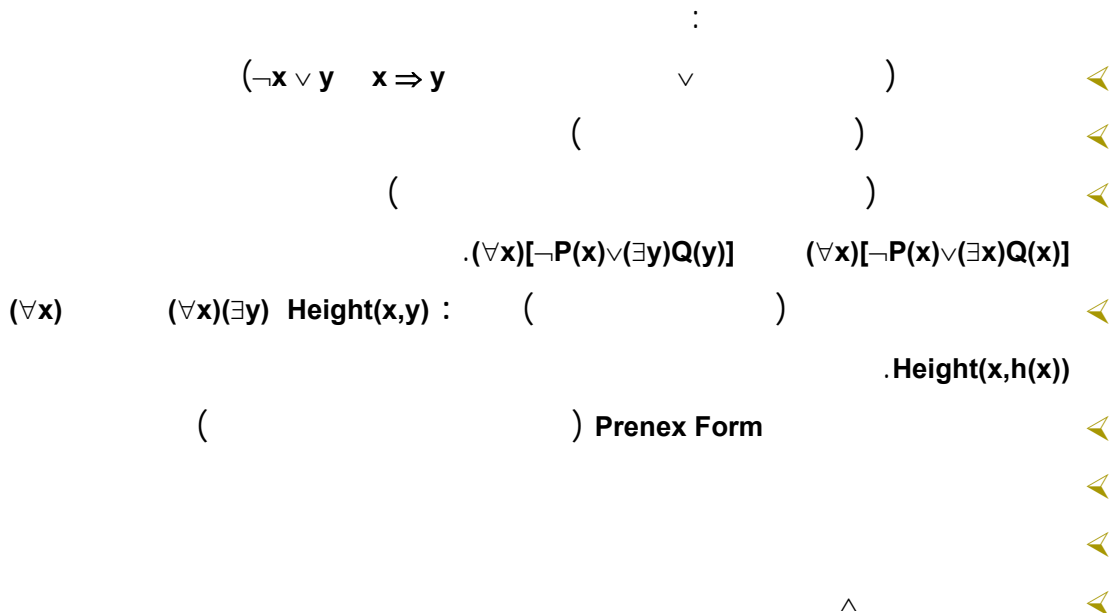
إذا وجد عنصران v_k و t_k في المجموعة D_k حيث v_k متغير لا يظهر في t_k تابع، وإلا فأخرج وأعلن الإخفاق، وتكون Γ غير قابلة للتوحيد

لاحظ أن $\Gamma_{k+1} = \Gamma_k \sigma_{k+1}$

أضف واحداً إلى k

أذهب إلى الخطوة 2

wff





27

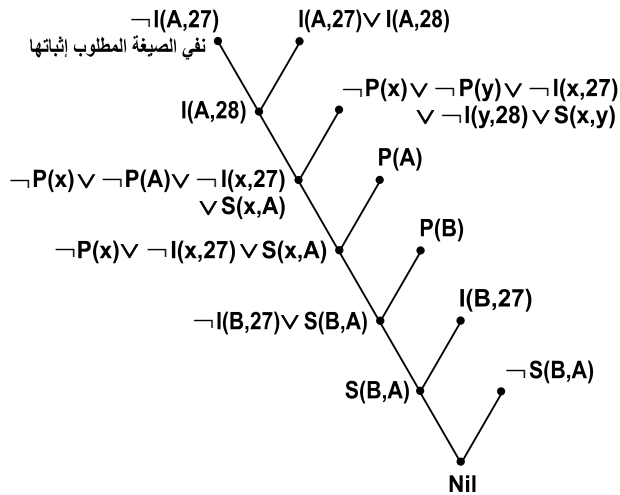
: .28

$\forall(x,y)\{\text{package}(x)\wedge\text{package}(y)\wedge\text{inroom}(x,27)\wedge\text{inroom}(y,28)\}\supset\text{smaller}(x,y)$

$\neg P(x)\neg\forall P(y)\neg\forall I(x,27)\neg\forall$

$I(y,28)\vee S(x,y)$

P = package, I = inroom, S = smaller :



الحل بالنقض

	I(B,27)	I(A,27) ∨	I(A,28)	P(A),P(B)
				¬S(B,A)
	27		A	B A
	B 27		B	28
				A
	27		A	:

Ans (u)

(A

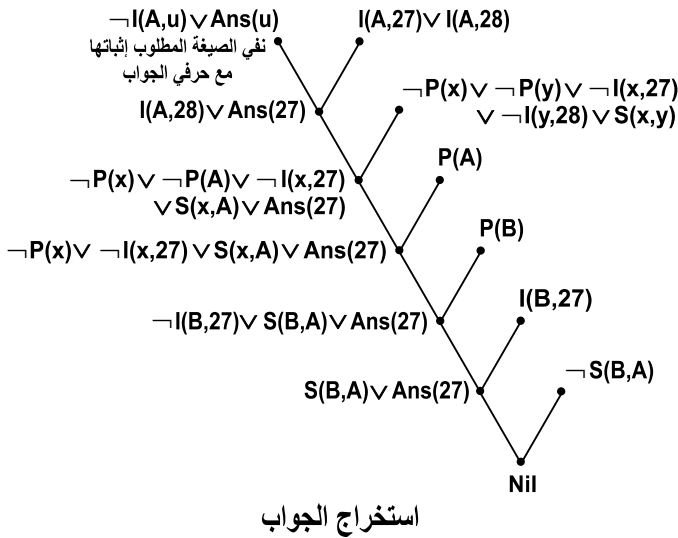
)

u

u

(∃u) I(A,u) :

.A



.A 27

.4

.1

P(A,y,z) P(x,B,B) .1

P(x,x) P(g(f(v)),g(u)) .2

P(y,y) (x,f(x)) .3

P(z,x,z) P(y,y,B) .4

x=3+3 2+3=x .5

.P(f(y,f(y,A)),A)

P(f(x,x),A)

.2

: clause form

.3

((∃-x)[P(x)] ∨ (∃x)[Q(x)]) ⊃ (∃x)[P(x) ∨ Q(x)] .1

(∀x)[P(x) ⊃ (∀y)[(∀z)[Q(x,y)] ⊃ ¬(∀z)[R(y,x)]]] .2

(∀x)[P(x) ⊃ (∃x)[(∀z)[Q(x,z)] ∨ (∀z)[R(x,y,z)]]] .3

(∀x)[P(x) ⊃ Q(x,y)] ⊃ ((∃y)[P(y)] ∧ (∃z)[Q(y,z)]) .4

.4

"Tony, Mike, and John belong to the Alpine Club. Every member of the Alpine Club is either a skier or a mountain climber or both. No mountain climber likes rain, and all skiers like snow. Mike dislikes whatever Tony likes and likes whatever Tony dislikes. Tony likes rain and snow."

	.Alpine	John	Mike	Tony"
.Tony	Tony	Mike	.	"
				Tony

"Who is a member of the Alpine Club who is a mountain climber but not a skier?"

:Oscar Clyde Sam

.5

1. Sam is pink.
2. Clyde is gray and likes Oscar.
3. Oscar is either pink or gray (but not both) and likes Sam.

			Sam	.1
		.Oscar	Clyde	.2
.Sam	()	Oscar	.3

$(\exists x,y)[\text{Gray}(x) \wedge \text{Pink}(y) \wedge \text{Likes}(x,y)]$

.5

2004

"

"



➤ Giarratano & Riley "Expert Systems: Principles and Programming", 3d Edition, 1998

.1

:
 .
 ()
 .
 ()

Prolog

Clips

()
 ()

Forward chainig

Backward Chaining

:
 (Q P) .Fact
 .(Rule)
 (P^Q ⊃ R) .

wff

) .Goal

(P^Q ⊃

:
 elephant(x) ⊃ mammel(x) : x x
 :

$mammal(x) \supset animal(x) : \quad x \quad x$
 $elephant(clide) : \quad Clide$
 $clide \quad x$
 $clide \quad x$
 $mammal(clide) : \quad clide \quad mammel(clide)$
 $clide \quad animal(clide)$
 $animal(clide)$
 $elephant(x) \supset mammel(x) : \quad x \quad x$
 $mammel(x) \supset animal(x) : \quad x \quad x$
 $animal(clide) : \quad clide$
 $animal(clide)$
 $mammel(clide)$
 $mammel(clide)$
 $elephant(clide)$
 $animal(clide)$

السلسلة الخلفية	السلسلة الأمامية	
التشخيص	التحكم، التخطيط	الاستخدام
الهدف	المعطيات	مقادير
من الأعلى إلى الأسفل، إيجاد حقائق تدعم فرضيات	من الأسفل إلى الأعلى، إيجاد النتائج التي تدعمها المعطيات	المحاكمة
عمقاً أولاً	عرضاً أولاً	يشبه خوارزميات البحث



⋮



.()

⋮

()

!

Doug Lenat

CYC

.2

)
:
!"
(
!

:



CYC

CYC .() Thing

:things

: ()

" X is a P, all P's are Q's, All Q's are R's"

"R Q Q P P X"

.frames

Snoopy :

Laser_Printer(Snoopy)

$(\forall x) [Laser_Printer(x) \supset Printer(x)]$

$(\forall x) [Printer(x) \supset Office_Machine(x)]$

categories Office_Machine Printer Laser_Printer



:

Office_Machine(Snoopy) $(\forall x) [Laser_Printer(x) \supset Office_Machine(x)]$



:

$(\forall x)[Office_Machine(x) \supset [Energy_Source(x) = Wall_Outlet]]$

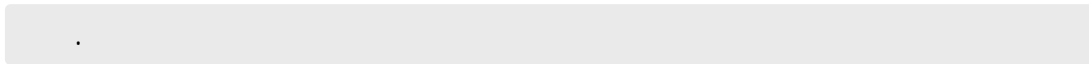
:



$(\forall x)[Laser_Printer(x) \supset [Energy_Source(x) = Wall_Outlet]]$

.Semantic network

:



:



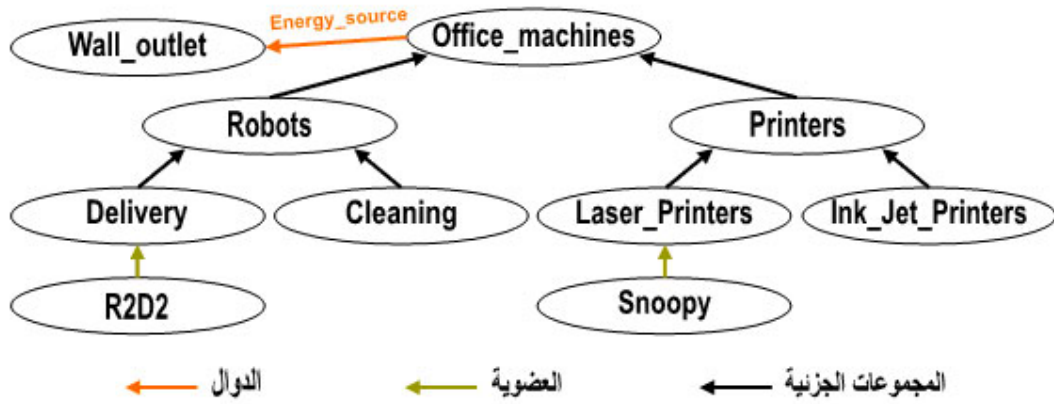
:

(isa isa links)

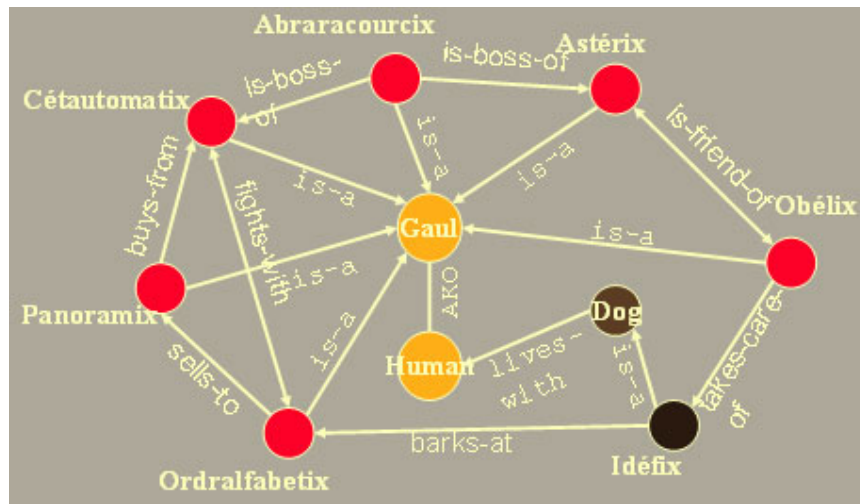


() ()

(1)



(2)



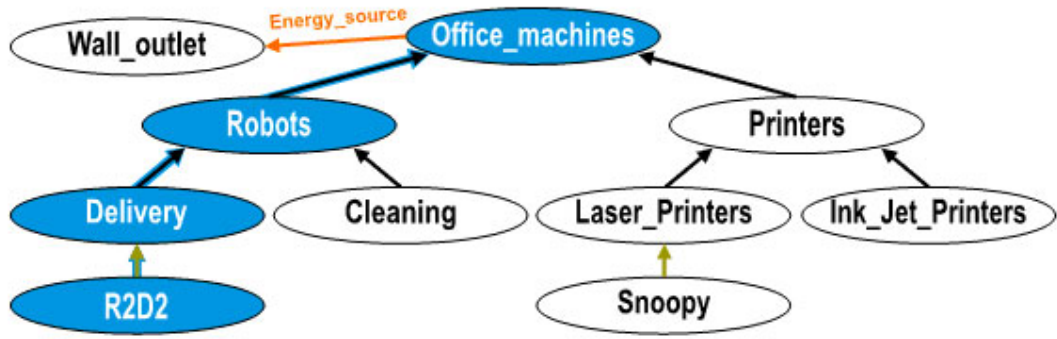
B

:A

(B) B : A

Office_Machine

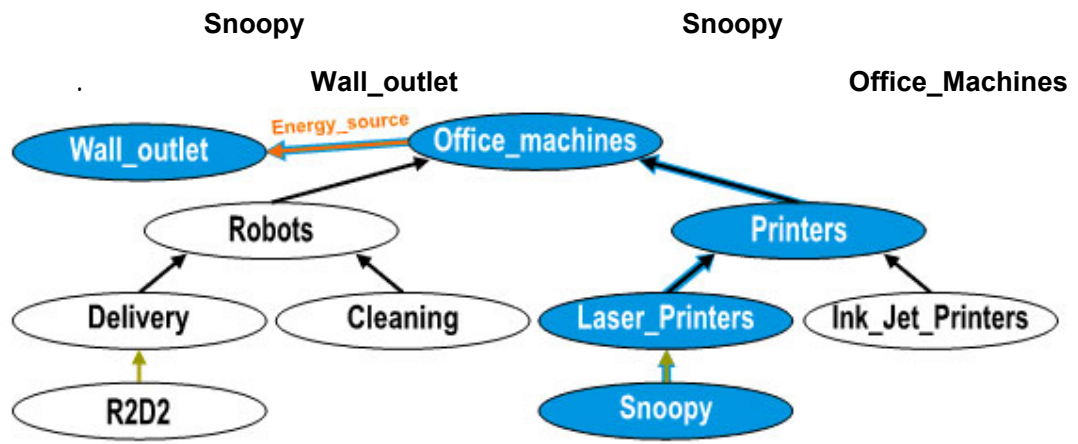
R2D2



← الدوال ← العضوية ← المجموعات الجزئية

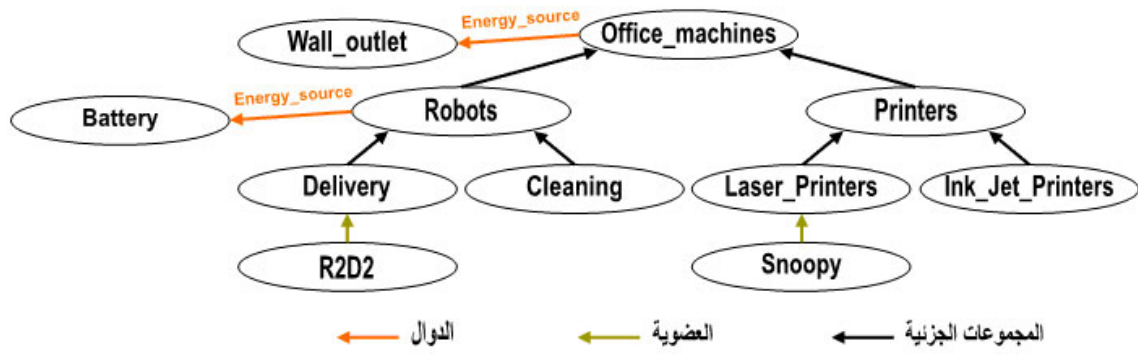
A

A



← الدوال ← العضوية ← المجموعات الجزئية

by default



()
)
(
)
(
: R2D2 R2D2 R2D2

Office_machines Energy_source Wall_outlet

.Battery

Robots

Energy_source

الغرض	الوصفة	قيمة الوصفة
Astérix	المهنة	محارب
Obélix	الحجم	كبير جداً
Idéfix	الحجم	صغير
Panoramix	الحكمة	لا نهائية

OAV (- -)

.heuristics ()

part-of

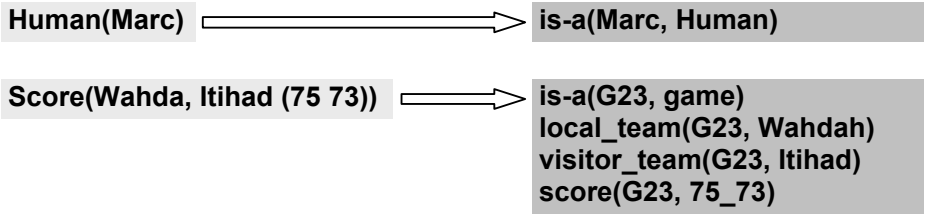
is-a

AKO

a-kind-of

:

:

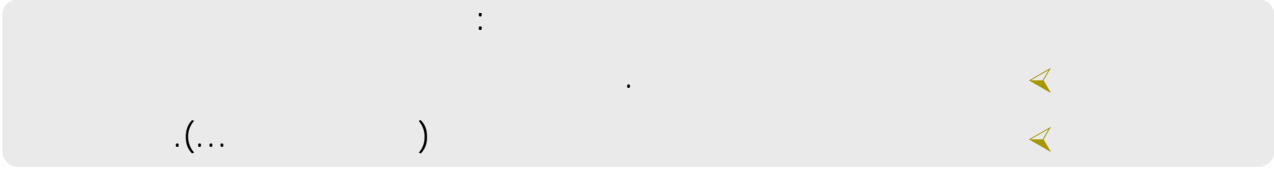


“Ahmad gave the book to Mona”

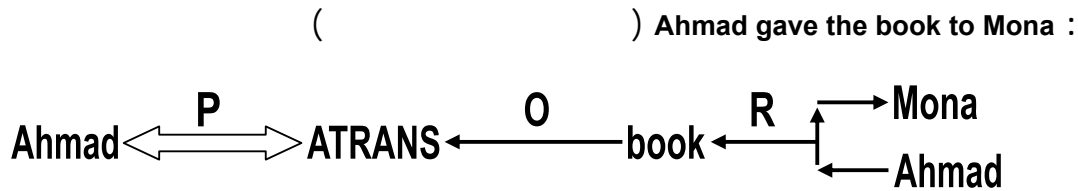
```
is_a(Action, give);
Tense(Action, past);
Subject(Action, Ahmad)
Obj1(Action, Mona)
Obj2(Action, book).
```

.3

Conceptual Dependencies :



Schank



Primitives

:ATRANS ●

:P ●

.receiver

:R ●

.object

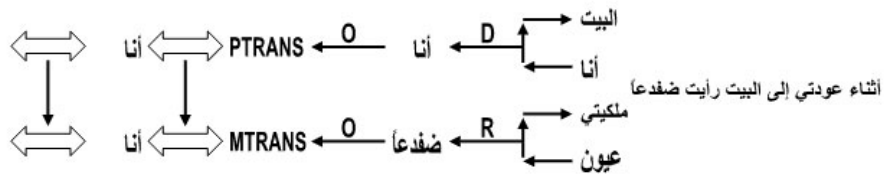
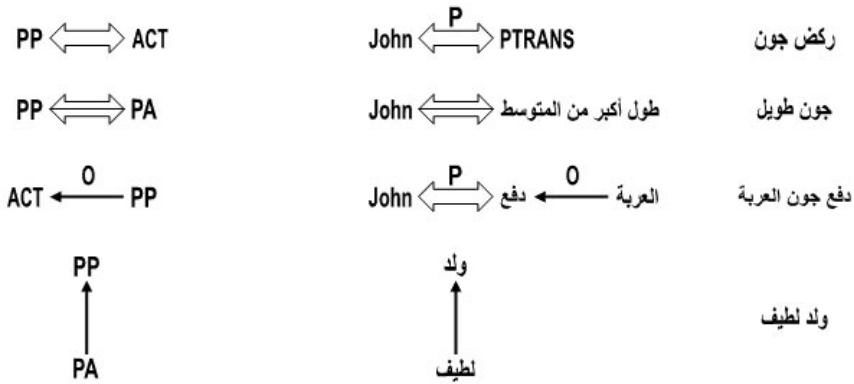
:O ●

Arabic Natural Language Processing

(www.ictis07.org :) ictis'07

April 2007

			:		
steal	take	give	:	Abstract Transfer	:ATRANS ●
				...	
	go		:	Physical Transfer	:PTRANS ●
				...	
	...	push	:		:PROPEL ●
	...	kick	:		:MOVE ●
		...	throw	:	:GRASP ●
		...		eat	:INGEST ●
		...		cry	:EXPEL ●
...			say	:	:MTRANS ●
	...	decide	:		:MBUILD ●
		...	sing,	:	:SPEAK ●
	...		listen		:ATTEND ●
			:		
(...)		Actions	:ACT ●
(..)		Picture Producers	:PP ●
(..)				Action Assistant	:AA ●
(..)				Picture Assistant	:PA ●
				:	



(...)

.ATRANS

t: تحول أو انتقال
 K: مستمر Continuous
 D أو Delta: دائم eternal

f: المستقبل future
 tf: تحول منته finished transition
 nil: الحاضر
 c: شرطي conditional

P: الماضي past
 ts: التحول البدئي start transition
 ?: استفهامي
 !: نفي

record

```

:
( slots' filling )
( )

```

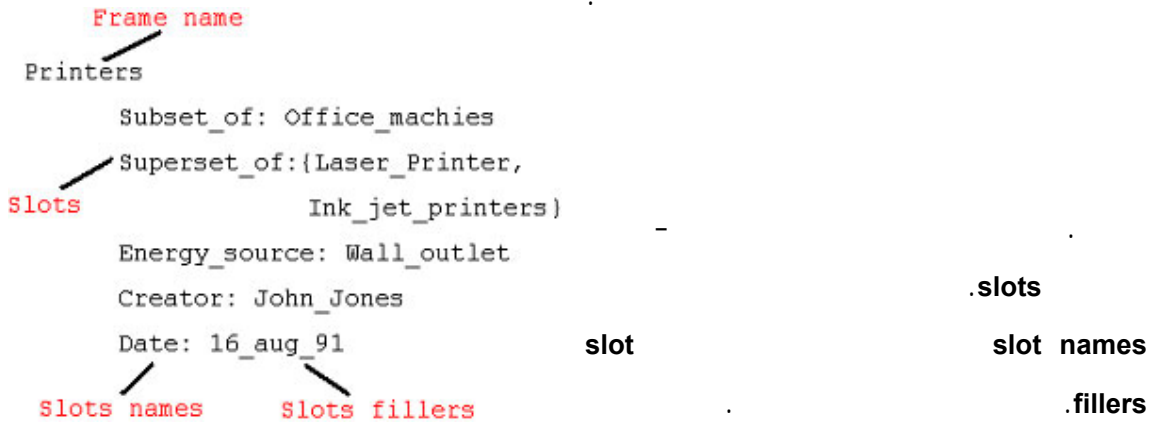
Frames .4

```

meta- )
:1975 Marvin Minsky (knowledge

```

"When one encounters a new situation (or makes a substantial change in one's view of a problem) one selects from memory a structure called a 'frame'. This is a remembered framework to be adapted to fit reality by changing details as necessary."



default values

meta knowledge

```

)
:(

```


Quantas Boarding Pass	Air New Zealand Boarding Pass
الشركة: Quantas	الشركة : Air New Zealand
اسم المسافر: Mr. Black	اسم المسافر : Mr. White
الرحلة: Q 101	الرحلة : NZ 101
التاريخ: 12 Dec	التاريخ : 13 Dec
المقعد: 24A	المقعد : 25A
من: Melbourne	من : Melbourne
إلى: Sydney	إلى : Christchurch
ساعة ركوب الطائرة: 0600	ساعة ركوب الطائرة : 1800
بوابة: 4	بوابة : 7

-
-
-
-
-
-
-
-
-
-


CLASS: Boarding Pass
 [Str]: شركة النقل
 [Str]: اسم المسافر
 [Str]: الرحلة
 [Str]: التاريخ
 [Str]: المقعد
 [Str]: من
 [Str]: إلى
 [Num]: ساعة ركوب الطائرة
 [Num]: بوابة


:
:Class-Frame <
 Class of computers .
:Instance-Frame () <
 My_computer . <


:
 :
 :
.is-a <
 :) <
 .(<

generic 

:


.(:) **is-a, a-kind-of** :Generalization 

.(:) **part-of, whole-part** :Aggregation 


X :) :Association 

.(**X**

:

() **inherent** : 


.()


:!demon **methods** 

demon


:


IF-THEN


when-needed 


when-changed 


:

.goals 


demon 


.bottom-up 

.top-down 

(meta knowledge) 

.Backward Chaining

"when needed" 

:) 

(. " "

:

	↖
()	↖
	↖
:	↖
	↖
	↖
	↖
.demons methods	↖
:	
	.1
	.2
	.3
	.4
.(when changed when-needed) demons	.5
	.6
	.7
Buy Smart :	
() :	↖
:	
	•
	•
	•
	•
	•
	•
:	
	•

(..)

-
-
-
-

Property

-
-
-

(...)

CLASS: Property
[Str] <i>Area:</i>
[Str] <i>Suburb:</i>
[N] <i>Price:</i>
[Str] <i>Type:</i>
[N] <i>Bedrooms:</i>
[N] <i>Bathrooms:</i>
[Str] <i>Construction:</i>
[Str] <i>Phone:</i>
[Str] <i>Pictfile:</i>
[Str] <i>Textfile:</i>
[N] <i>Instance Number:</i>

.Instances



INSTANCE: Property 1
<i>Class:</i> Property
[Str] <i>Area:</i> دمشق
[Str] <i>Suburb:</i> برزة
[N] <i>Price:</i> 1640000
[Str] <i>Type:</i> شقة
[N] <i>Bedrooms:</i> 3
[N] <i>Bathrooms:</i> 1
[Str] <i>Construction:</i> بناء حديث
[Str] <i>Phone:</i> (03) 6226 4212
[Str] <i>Pictfile:</i> house01.bmp
[Str] <i>Textfile:</i> house01.txt
[N] <i>Instance Number:</i> 1

INSTANCE: Property 2
<i>Class:</i> Property
[Str] <i>Area:</i> دمشق
[Str] <i>Suburb:</i> مهاجرين
[N] <i>Price:</i> 1500000
[Str] <i>Type:</i> شقة
[N] <i>Bedrooms:</i> 3
[N] <i>Bathrooms:</i> 1
[Str] <i>Construction:</i> قديم
[Str] <i>Phone:</i> (03) 6226 1416
[Str] <i>Pictfile:</i> house02.bmp
[Str] <i>Textfile:</i> house02.txt
[N] <i>Instance Number:</i> 2

.Display





When needed & When :demon methods

.changed

.when needed & when changed

.pattern matching

demon

.context

scenario script

by-default

slots

(Schank, 1977)

:Script

F : ●

. O M C W S : ●

S S : ●

() S S O S : ●

<p>Script: RESTAURANT Track: Coffee Shop Props: Tables Menu F = Food Check Money</p> <p>Roles: S = Customer W = Waiter C = Cook M = Cashier O = Owner</p>	<p>Scene 1: Entering</p> <p>S PTRANS S into restaurant S ATTEND eyes to tables S MBUILD where to sit S PTRANS S to table S MOVE S to sitting position</p>
<p>Entry conditions: S is hungry. S has money.</p> <p>Results: S has less money O has more money S is not hungry S is pleased (optional)</p>	<p>Scene 2: Ordering</p> <p>(Menu on table) (W brings menu) (S asks for menu) S PTRANS menu to S S MTRANS signal to W W PTRANS W to table W PTRANS W to table W ATRANS menu to S S MTRANS "need menu" to W W PTRANS W to menu</p> <p>S MTRANS food list to CP(S) ^S MBUILD choice of F S MTRANS signal to W W PTRANS W to table S MTRANS "I want F" to W</p> <p>W PTRANS W to C W MTRANS (ATRANS) to C</p> <p>C MTRANS "no F" to W C DO (prepare F script) W PTRANS W to S to Scene 3 W MTRANS "no F" to S (go back to ") or (go to Scene 4 at no pay path)</p>
	<p>Scene 3: Eating</p> <p>C ATRANS F to W W ATRANS F to S S INGEST F</p> <p>(Option: Return to Scene 2 to order more; Otherwise, go to Scene 4)</p>
	<p>Scene 4: Existing</p> <p>S MTRANS to W (W ATRANS check to S</p> <p>W MOVE (write check) W PTRANS W to S W ATRANS check to S S ATRANS tip to W S PTRANS S to M S ATRANS money to M</p> <p>(No pay path) S PTRANS S to out of restaurant</p>

Generic RESTAURANT Frame

Specialization-of: Business-Establishment

Types:

range: (Cafeteria, Fast-Food, Seat-Yourself, Wait-To-Be-Seated)

default: Seat-Yourself

if-needed: IF plastic-orange-counter THEN Fast-Food,
IF stack-of-trays THEN Cafeteria,
IF wait-for-waitress-sign or reservations-made THEN Wait-To-Be-Seated,
OTHERWISE Seat-Yourself.

Location:

range: an ADDRESS

if-needed: (Look at the MENU)

Name:

if-needed: (Look at the MENU)

Food-Style:

range: (Burgers, Chinese, American, Seafood, French)

default: American

if-added: (Update Alternatives of Restaurant)

Times-of-Operation:

range: a Time-of-Day

default: open evenings except Mondays

Payment-Form:

range: (Cash, CreditCard, Check, Washing-Dishes-Script)

Event-Sequence:

default: Eat-at-Restaurant Script

Alternatives:

range: all restaurants with same Foodstyle

if-needed: (Find all Restaurants with the same Foodstyle)

[Rogers 1999]

()

EAT-AT-RESTAURANT Script

Props: (Restaurant, Money, Food, Menu, Tables, Chairs)

Roles: (Hungry-Persons, Wait-Persons, Chef-Persons)

Point-of-View: Hungry-Persons

Time-of-Occurrence: (Times-of-Operation of Restaurant)

Place-of-Occurrence: (Location of Restaurant)

Event-Sequence:

first: Enter-Restaurant Script

then: if (Wait-To-Be-Seated-Sign or Reservations)
then Get-Maitre-d's-Attention Script

then: Please-Be-Seated Script

then: Order-Food-Script

then: Eat-Food-Script unless (Long-Wait) when Exit-Restaurant-Angry Script

then: if (Food-Quality was better than Palatable)
then Compliments-To-The-Chef Script

then: Pay-For-It-Script

finally: Leave-Restaurant Script

[Rogers 1999]

```

: .
( . )
" "
):" "
(
.
"
" "
):"
(
):
(
.
.6
" .1
"

```

: .1
ATRANS, PTRANS, MTRANS, DIGEST, MOVE, GRASP, EXPEL, ATTEND, SPEAK

.2
script
•
•
•
•
•
•

" :
 :
ATRANS, PTRANS, MTRANS, INGEST, MOVE, GRASP, EXPEL, ATTEND, SPEAK
script

:
 .()
 .3

()
 ()
instance

.7
 . 2007 – 2006 :
 www.ictis07.org ANLP
What is a Knowledge Representation

.1

()

C B A

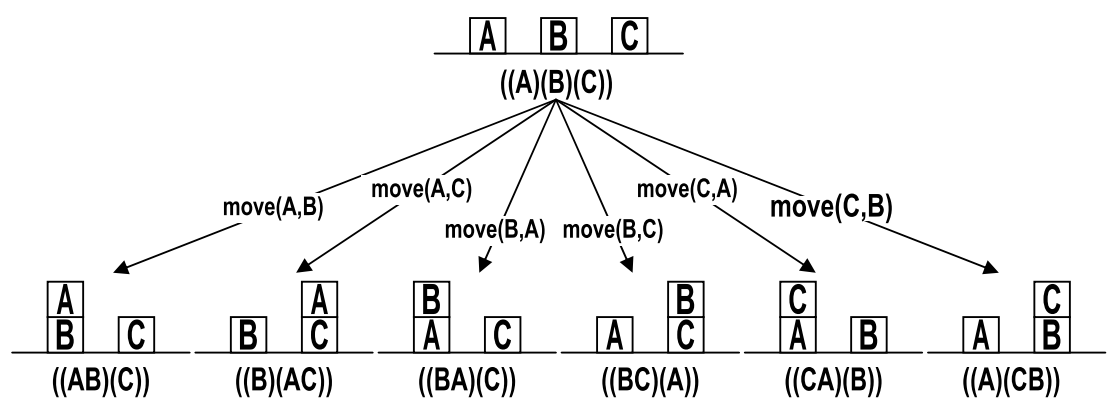
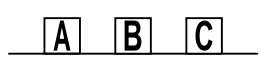
B B A

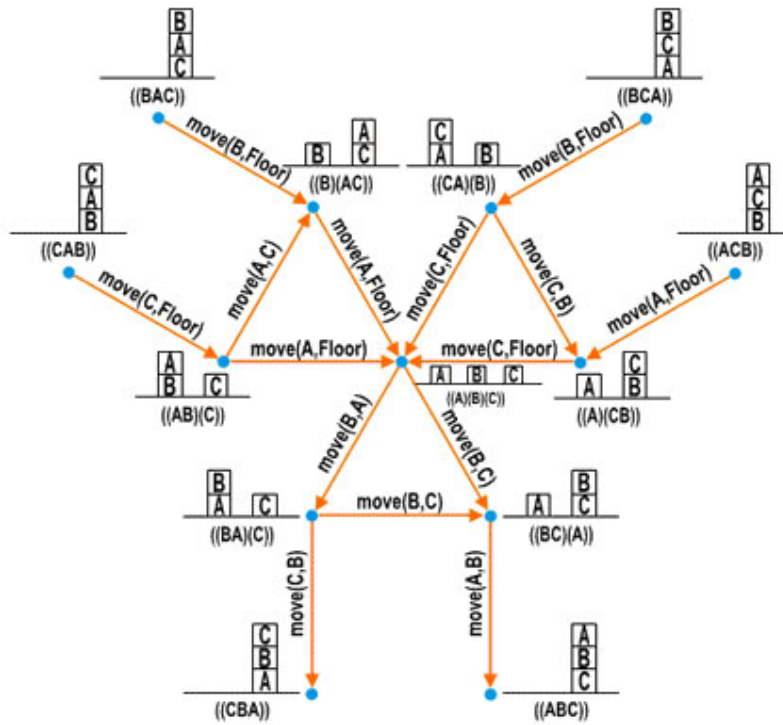
C C

x

y

move(x, y)





Graph

nodes

arcs

"directed graph"

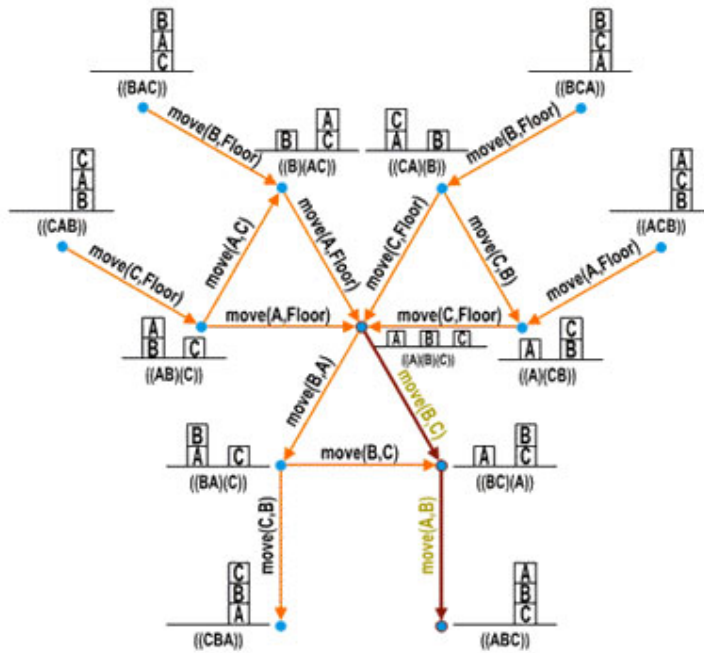
successor n_j n_j n_i

parent () n_i n_i (

((A)(B)(C)) :

((ABC))

{move(B,C),move(A,B)}



:

)

()

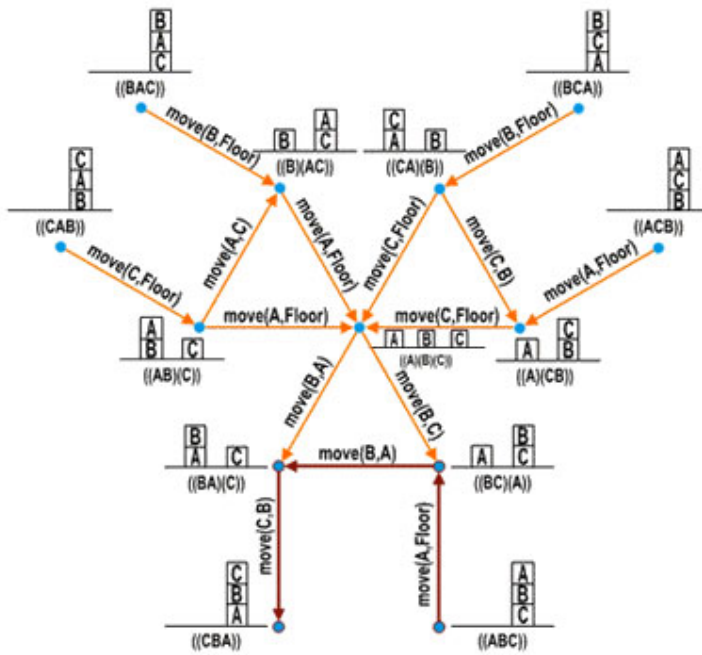
(

B A :

A B C

C

{move(A, Floor), move(B, A), move(C, B)}



)

(



عنوان العقد للبحث عن الهدف
(البحث الشامل)

"expansion"

) :

— (

.i < j j i

.()

() ()

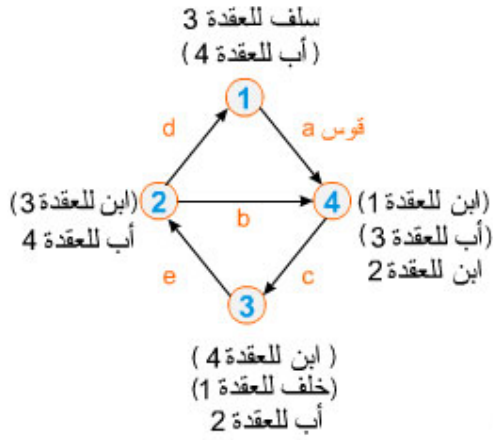
.root

depth

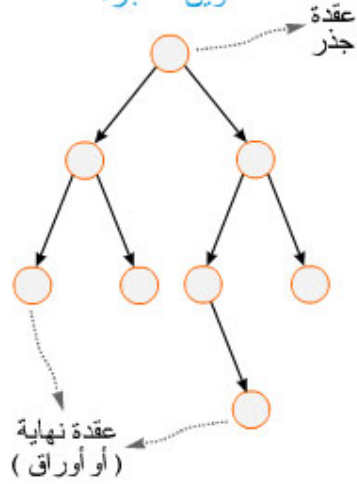
() .1

()

تكوين البيان



تكوين الشجرة



.uninformed

.heuristic ()

heuristic

heuristic()

("! ") Eureka!

Uninformed

:Open

:Close

()

.Successor function

expanding :

FIFO

Open

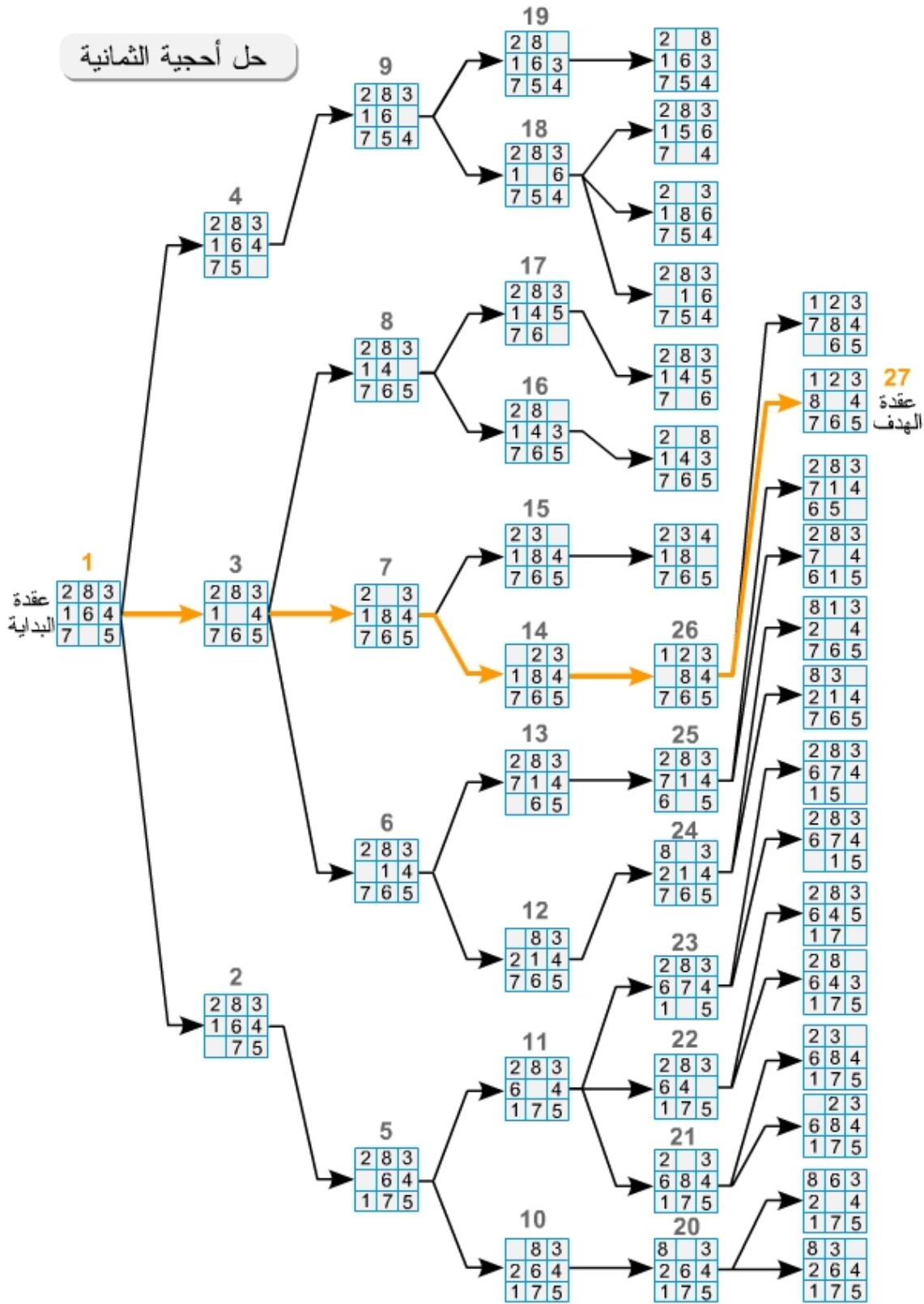
—

" "

(1)

()

حل أحجية الثمانية



depth bound

.() .

LIFO

Open

" _ "

()

.3

n

<http://www.mazeworks.com/hanoi>

n-1 .1

.2

n-1 .3





.and-or

:



and-or .4

M1 or M2

:

or M3

M1: OP11 and OP12 and OP13
M2: OP21 and OP22 and OP23
M3: OP31 and OP32 and OP33

.and-or

:

and-or



(and)

or

and-or

and



.() .

and-or

:

or



.() and

or

and



.(fathers)

or

()

.5

$$\hat{f}(n) \quad .1$$

$$\hat{f}(n) \quad .2$$

$$. (\quad) \quad .3$$

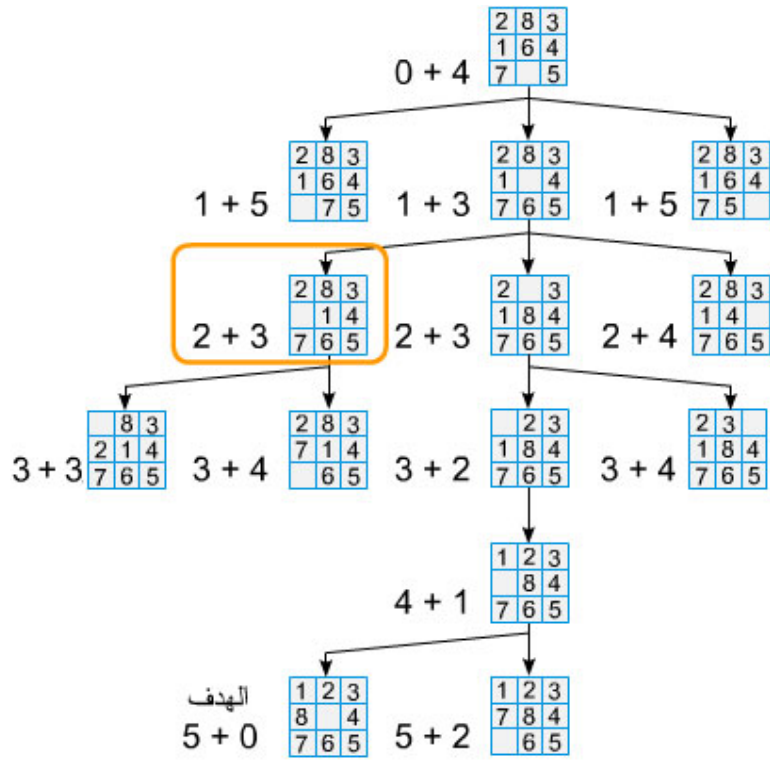
$$\hat{f}(n) = (\quad)$$

$$\hat{f}(n) = \hat{g}(n) + \hat{h}(n)$$

$$(n \quad) \quad n \quad \hat{g}(n) \quad \hat{h}(n)$$

$$\hat{h}(n) = (\quad)$$

$$\hat{g}(n) = \quad n$$



graphsearch

1. n_0 . n_0 Tr . OPEN
 2. CLOSED .2
 3. OPEN .3
 4. OPEN OPEN .4
 5. n .5
 6. Tr n M M n .6
 7. OPEN M .M n .7
 8. .3 .8
-) OPEN
- ((LIFO) OPEN (FIFO)

OPEN

A*

$f = h + g$

n_0

\hat{f}

n

f

$.n$

$:n$

$.(\$

$)n$

n_0

$:\hat{g}(n)$

n

$:\hat{h}(n)$

$h(n) \quad g(n)$

h

g

$.g$

f

.6



:

Q P : P∨Q

.Q P how certain

.P⊃Q P Q

()

.1

joint

.V₁, V₂, ..., V_k

p(V₁=v₁,

v₁, v₂, ..., v_k

V₁, V₂, ..., V_k

probability

.V₂=v₂, ..., V_k=v_k)

V₁,

joint probability function

p(V₁,V₂, ...,V_k)

.V₂,..., V_k

Head (H)

)

.p(H)=1/2

(Tail (T)

:

p(H , T, T, H, T)=1/32

"H "

p(H , T, T, H, T)

."T "

"H "

"T "

"T "

:

0 ≤ p(V₁, V₂,..., V_k) ≤ 1

(a)

∑ p(V₁, V₂,..., V_k) = 1

(b)

p(H)=1/2

(b)

(a)

.p(T)=1/2

()

.False True

.False True

()

LIFTABLE MOVES BAT_OK

()

GAUGE

.G L M B

.False True

.True

False M

16

p(B=b, M=m, L=l, G=g)

.False True g l m b

(B, M, L, G)	Joint probability (الاحتمال المشترك)
(True, True, True, True)	0.5685
(True, True, True, False)	0.0299
(True, True, False, True)	0.0135
(True, True, False, False)	0.0007
(..., ..., ..., ...)	...

)

(

marginal probability

p(B=b)

.B=b

$$p(B = b) = \sum_{B=b} p(B, M, L, G)$$

$$p(B=True)=0.95$$

. True B

$$p(B=b, M=m)$$

: M=m B=b

$$p(B = b, M = m) = \sum_{B=b, M=m} p(B, M, L, G)$$

:

$$p(B = b, M = m) = \sum_{B=b, M=m} p(B, M, L)$$

$$p(B = b) = \sum_{B=b} p(B, M)$$

(False True)

. p(B, ¬M)

p(B=True, M=False)

()

()

Probabilistic Inference

V_i) $p(V_i|V_j)$ V_j V_i

: (V_j)

$$p(V_i|V_j) = p(V_i, V_j) p(V_j)$$

. V_j

$p(V_j)$ V_j V_i

$p(V_i, V_j)$

:

$$p(V_i, V_j) = p(V_i|V_j) p(V_j)$$

:

$$p(B = True | M = False) = p(B = True, M = False) / p(M = False)$$

$p(M = False)$

$p(B = True, M = False)$

.frequency

p(M = False)

.()

$$p(B) = p(B,M) + p(B,-M)$$

:()

$$p(-G,B-M,L) = p(-G,B,-M,L) / p(-M,L)$$

: chain

$$p(B,L,G,M) = p(B|L,G,M) p(L|G,M) p(G|M) p(M)$$

)

:()

$$p(V_i, V_j) = p(V_i|V_j) p(V_j) = p(V_j|V_i) p(V_i) = p(V_j, V_i)$$

or

$$p(V_i|V_j) = p(V_j | V_i) p(V_i) / p(V_j)$$

.Bayes' Rule

Reverend Thomas Bayes [Bayes 1763]

.2

$$P(A/B) = P(B/A) \cdot P(A) / P(B)$$

:

.A P(A)

.B P(B)

P(B/A) P(A/B)

B A

: -B B

A

P(Event/

:Posterior Prob.

.Evidence)

.E_j ()

H_i

: .P(H_i/E_j)

P(E_j/H_i)

P(H_i)

$$p(H | E) = \frac{p(E | H).p(H)}{p(E | H).p(H) + p(E | \neg H).p(\neg H)}$$

p(H), p(¬H)

.() .

p(H | E)

p(E | H), p(E | ¬H)

E_j, j=1,...,n

n

H_i, i=1,...,m

m

:

$$p(H_i | E_1, \dots, E_n) = \frac{p(E_1, \dots, E_n | H_i).p(H_i)}{p(E_1, \dots, E_n)}$$

$$p(H_i | E_1, \dots, E_n) = \frac{p(E_1 | H_i).p(E_2 | H_i) \dots p(E_n | H_i).p(H_i)}{\sum_j p(E_1 | H_j).p(E_2 | H_j) \dots p(E_n | H_j).p(H_j)}$$

:

:H₁

:H₂

:H₃

:

	Hypothesis		
	i = 1	i = 2	i = 3
P(H _i)	.70	.10	.20
P(E ₁ H _i)	.80	.50	.10
P(E ₂ H _i)	.30	.50	.70
P(E ₃ H _i)	.70	.30	.05

:E₁

3 :E₂

:E₃

:

P(H₁/E₁) <

$$P(H_1/E_1) = P(E_1/H_1).P(H_1) / PP$$

:

$$PP = P(E_1/H_1). P(H_1) + P(E_1/H_2). P(H_2) + P(E_1/H_3). P(H_3)$$

:

$$PP = .7*.8 + .1*.5 + .2*.1 = .63$$

3

$$P(H_3/E_2) P(H_1/E_1) = .56/.63 = .889 \leftarrow$$

$$P(H_3/E_2) = P(E_2/H_3) \cdot P(H_3) / PP_1$$

$$PP_1 = P(E_2/H_1) \cdot P(H_1) + P(E_2/H_2) \cdot P(H_2) + P(E_2/H_3) \cdot P(H_3)$$

$$PP_1 = .7 \cdot .3 + .1 \cdot .5 + .2 \cdot .7 = .4$$

$$P(H_1/E_1, E_3) P(H_3/E_2) = .14/.40 = .35 \leftarrow$$

$$P(H_1/E_1, E_2) = P(E_1/H_1) \cdot P(E_3/H_1) \cdot P(H_1) / PP_2$$

$$PP_2 = P(E_1/H_1) \cdot P(E_3/H_1) \cdot P(H_1) + P(E_1/H_2) \cdot P(E_3/H_2) \cdot P(H_2) + P(E_1/H_3) \cdot P(E_3/H_3) \cdot P(H_3)$$

$$PP_2 = .392 + .015 + .001$$

$$P(H_1/E_1, E_3) = .392 / PP_2 = .961$$

3

:

:



:



.3

.PROSPCTOR MYCIN

Ray Simpson (1944)		Milton Hikel (1968)	
Term	Mean value	Term	Mean value
Always	99	Always	100
Very often	88	Very often	87
Usually	85	Usually	79
Often	78	Often	74
Generally	78	Generally	74
Frequently	73	Frequently	72
Rather often	65	Rather often	72
About as often as not	50	About as often as not	50
Now and then	20	Now and then	34
Sometimes	20	Sometimes	29
Occasionally	20	Occasionally	28
Once in a while	15	Once in a while	22
Not often	13	Not often	16
Usually not	10	Usually not	16
Seldom	10	Seldom	9
Hardly ever	7	Hardly ever	8
Very seldom	6	Very seldom	7
Rarely	5	Rarely	5
Almost never	3	Almost never	2
Never	0	Never	0

(IF part)

(THEN part)

(...)

()

()

()

(..)

Certainty Factors

.4

Mycin

الرقم	دلالتة اللغوية
-1	أكد لا
-0.8	تقريباً أكد لا
-0.6	من المحتمل لا
-0.4	ربما لا
-0.2 to 0.2	غير معروف
+0.4	ربما نعم
+0.6	من المحتمل نعم
+0.8	تقريباً أكد نعم
+1	أكد نعم

+1

-1

0



)

(

AND

$$CF(A \text{ and } B) = \min (CF(A), CF(B))$$

OR

$$CF(A \text{ or } B) = \max (CF(A), CF(B))$$

$$CF(\neg A) = - CF(A)$$

: $CF(A) = 0.3$ $CF(B) = 0.5$ $CF(C) = 0.4$ $CF(D) = -0.7$:
 $CF(A \text{ and } B \text{ or } C \text{ and not}(D)) = \max(\min(0.3,0.5),\min(0.4,0.7))=\max(0.3,0.4)=0.4$

0.5

0.6

$$.0.6 \times 0.5 = 0.3$$

Q_1

$$P_2 \Rightarrow Q_1 \quad P_1 \Rightarrow Q_1$$

cf_2

Q_1

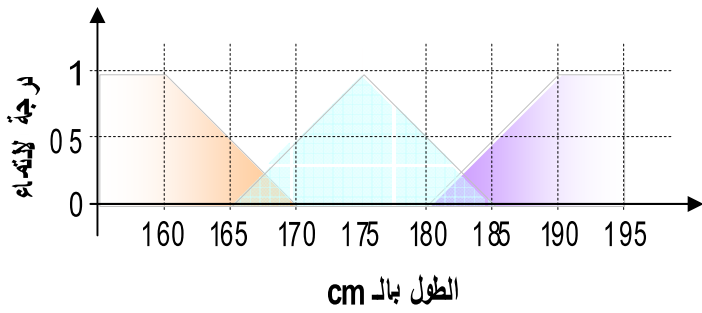
cf_1

:

Q_1

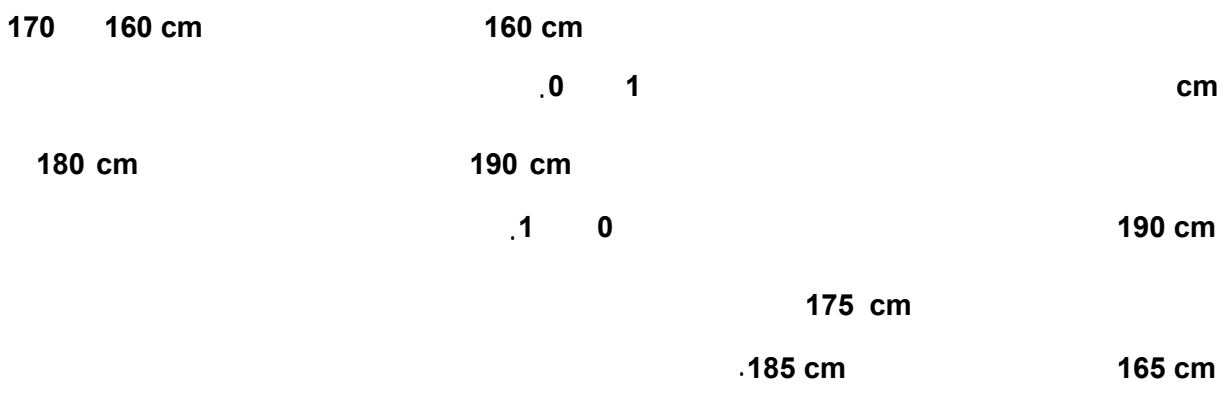
$$cf(cf_1, cf_2) = \begin{cases} cf_1 + cf_2 \times (1 - cf_1) & \text{if } cf_1 > 0 \text{ and } cf_2 > 0 \\ \frac{cf_1 + cf_2}{1 - \min[|cf_1|, |cf_2|]} & \text{if } cf_1 < 0 \text{ or } cf_2 < 0 \\ cf_1 + cf_2 \times (1 + cf_1) & \text{if } cf_1 < 0 \text{ and } cf_2 < 0 \end{cases}$$

.5



$x \in X$: crisp set
 $x \notin X$:
 " "

$\mu_A(x) : X \rightarrow [0,1]$
 $\mu_A(x) = 1$ if x is totally in A ;
 $\mu_A(x) = 0$ if x is not in A
 $0 < \mu_A(x) < 1$ if x is partly in A



()

.Negnevsky

.6

4

:



Michael

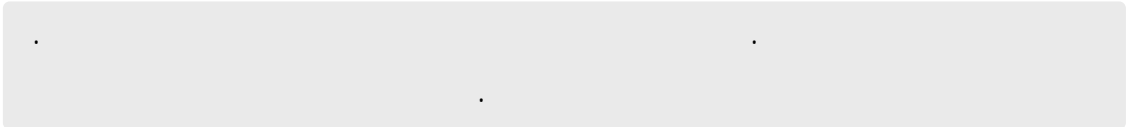
artificial Intelligence: a guide to intelligent systems



.2002

Addison Wisley

Negnevisky



()
:()

.meta rules

)
.(:



Inductive
Deductive ()



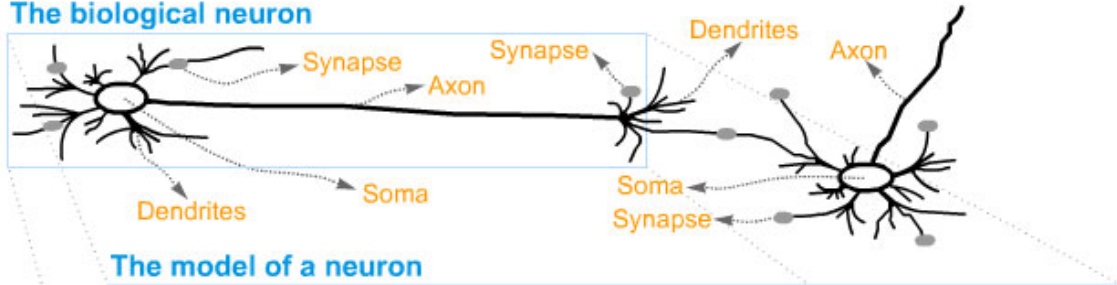
: .2

neural networks

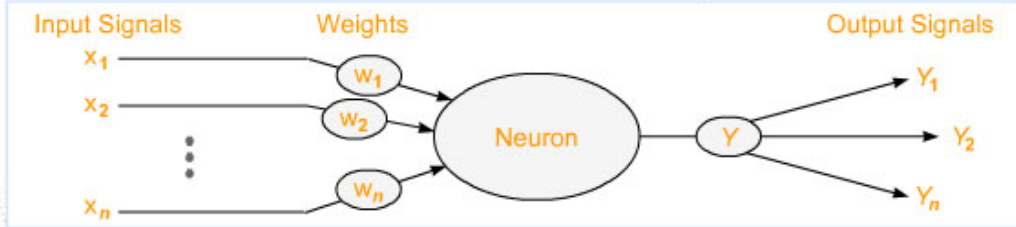
(
a ()
classes labels
f(X)
f
training set
supervised
X
n
X
n
x_i
i=1, ..., n
)

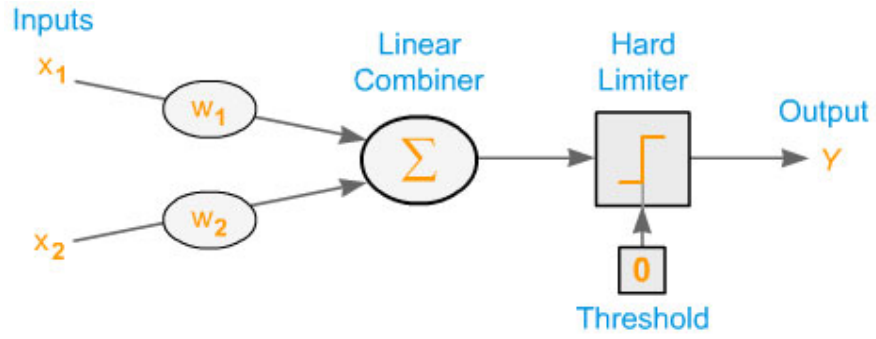
)
" " ()
" " f accuracy " "
.()
)
(
()
) supervised
.(
()
:

The biological neuron



The model of a neuron





()

OK	يجب الموافقة على القرض
COLLAT	ضمانة القرض الإضافية مقنعة
PYMT	طالب المال (العميل) قادر على سداد دفعات القرض
REP	للعامل سمعة مالية جيدة
APP	تخمين الضمانة أكبر من مبلغ القرض بقدر كافٍ
RATING	للعامل دفعات دورية منتظمة
INC	دخل العميل يتجاوز مصاريفه
BAL	للعامل نشرة موازنة ممتازة

.3

.Explanation-Based Generalization (EBG)

- 1. COLLAT \wedge PYMT \wedge REP \supset OK
- 2. APP \supset COLLAT
- 3. RATING \supset REP
- 4. INC \supset PYMT
- 5. BAL \wedge REP \supset OK

OK :

AND/OR

OK

OK ()

()

()

()

OK

.REP PYMT COLLAT

each

both

REP

BAL

OK

OK

OR

.AND/OR

OR

AND/OR

()

True

OK

False

positive instances

.negative instances

$$\alpha_1 \wedge \alpha_2 \wedge \dots \wedge \alpha_n \supset OK$$

{APP, RATING, INC, BAL}

α_i

True ()

cover

.specific

.more general

"greedy"

"Separate and Conquer"

()

العميل	APP	RATING	INC	BAL	OK
1	1	0	0	1	0
2	0	0	1	0	0
3	1	1	0	1	1
4	0	1	1	1	1
5	0	1	1	0	0
6	1	1	1	0	1
7	1	1	1	1	1
8	1	0	1	0	0
9	1	1	0	0	0

معطيات المصرف

(نستخدم 1 لـ True و 0 لـ False)

)

(

T \supset OK :

{APP, RATING, INC, BAL}

$$r_\alpha = n_\alpha^+ / n_\alpha$$

()

()

n_α

()

n_α^+

α

α

r_α : r_α

$r_{APP} = 3/6 = 0.5$
 $r_{RATING} = 4/6 = 0.667$
 $r_{INC} = 3/6 = 0.5$
 $r_{BAL} = 3/4 = 0.75$

$BAL \supset OK$: BAL

1 .7 4 3

r_α

: BAL

$r_{APP} = 2/3 = 0.667$ $r_{RATING} = 3/3 = 1.0$ $r_{INC} = 2/2 = 1.0$

) . RATING .INC RATING
 .(INC

$BAL \wedge RATNG \supset OK$

.6

العميل	APP	RATING	INC	BAL	OK
1	1	0	0	1	0
2	0	0	1	0	0
3	1	1	0	1	1
4	0	1	1	1	1
5	0	1	1	0	0
6	1	1	1	0	1
7	1	1	1	1	1
8	1	0	1	0	0
9	1	1	0	0	0

معطيات المصرف

(نستخدم 1 لـ True و 0 لـ False)

$T \supset OK$

.9 8 5 2 1

$r_{APP} = 1/4 = 0.25$
 $r_{RATING} = 1/3 = 0.33$
 $r_{INC} = 1/4 = 0.25$
 $r_{BAL} = 0/1 = 0.0$

.RATING \supset OK

rRATING

9 5

RATING \supset OK

:

:

RATING

APP \wedge RATING \supset OK

()

.9

:

r_{RATING} = 1/2 = 0.5

r_{INC} = 1/2 = 0.5

r_{BAL} = 0/0

APP \wedge RATING \wedge INC \supset OK

:

BAL \wedge RATNG \supset OK

APP \wedge RATING \wedge INC \supset OK

العميل	APP	RATING	INC	BAL	OK
1	1	0	0	1	0
2	0	0	1	0	0
5	0	1	1	0	0
6	1	1	1	0	1
8	1	0	1	0	0
9	1	1	0	0	0

المعطيات المختصرة

(نستخدم 1 لـ True و 0 لـ False)

()

()

:



)

(

:



()

)

"

"

(

()

"

"

.Overfitting

.pseudocode

.Generic Separate-and-Conquer Algorithm (GSCA)

```

    .γ
    )
    .(
    ) Γ (
    ) γ
    .Ξ
    :GSCA
    pseudocode
    Ξcur ← Ξ .1
    π ← ϕ .2
    .repeat .3
    Γ ← T .4
    ρ ← Γ ⊃ y .5
    .repeat .6
    .choose(α) α .7
    .(
    Γ ← Γ ∧ α .8
    ) ρ until .9
    .(π
    ρ π ← π .10
    ) Ξcur ← Ξcur - (π
    Ξcur
    ) π until .12
    )
    .(
    .4

```

() .1

.2

.3

)

.1970

Waterman

.(

:

.()

: .1

: .2

: ●

: ●

: ●

3 0 : .3

: .4

: .5

:

↖

()

↖

↖

:

●

)

:

●

() ()

.5

.1992

ESIMAG



"

:2

"



.1995

Michael

artificial Intelligence: a guide to intelligent systems



.2002

Addison Wisley

Negnevisky

.1

()

/ / .2

) .1

.(

) .2

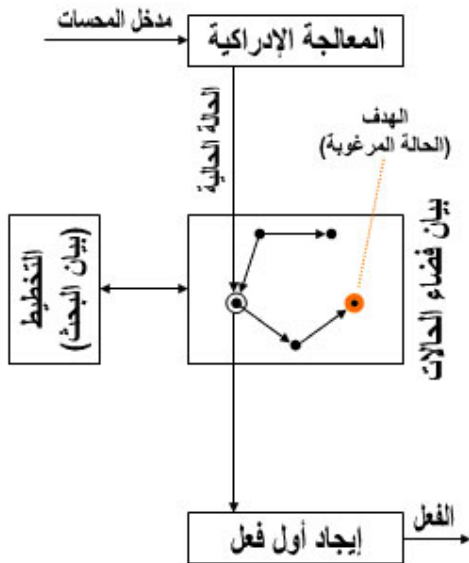
.(

.3

.4

.5

.6



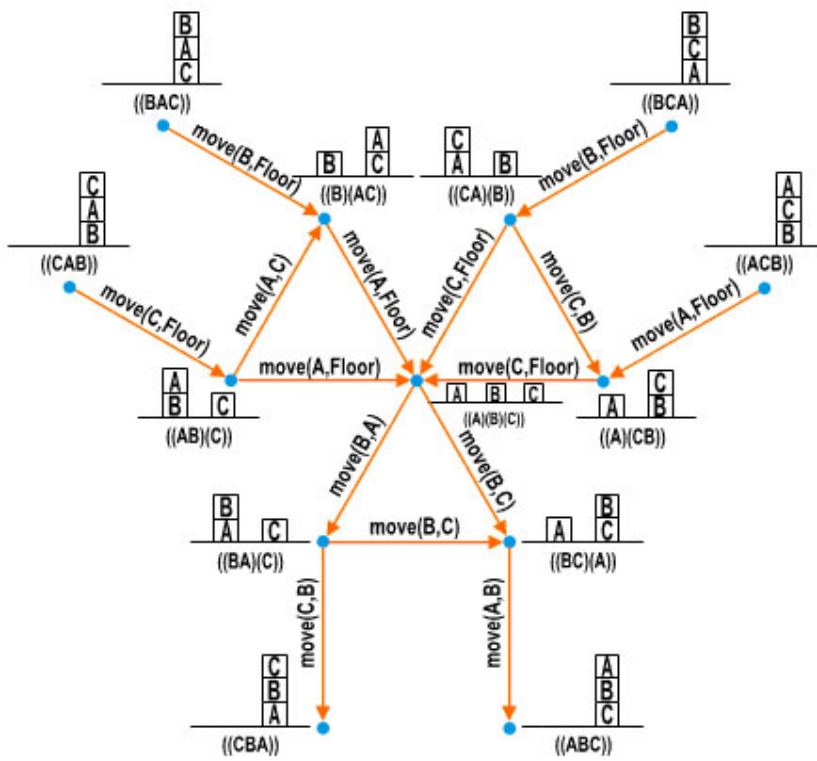
بنية وكيل تحسس / تخطيط / فعل

/ /

" () .()

/ /

/ /



.3

الأشجار الفرعية لمسألة تكديس الكتل

(CBA)

(BCA)

.A*

$$\hat{h}(n_i) \leftarrow \min_{n_j \in S(n_i)} [\hat{h}(n_j) + c(n_i, n_j)]$$

$$\begin{aligned} & \hat{h} \quad 0 \\ & : \quad \hat{h}(n_i) \quad S(n_i) \quad n_i \\ & \quad n_j \quad n_i \quad c(n_i, n_j) \\ & \quad \hat{h} \\ & \quad \hat{h}(n_g) = 0 \quad n_g \end{aligned}$$

\hat{h}

(!)

$$\hat{h}(n_i) \leftarrow [\hat{h}(n_j) + c(n_i, n_j)]$$

$c(n_i, n_j)$

$$\begin{aligned} & \hat{h} \\ & n_j \quad a \quad n_i \\ & n_j \quad 0 \quad n_j \quad \hat{h}(n_j) \\ & n \end{aligned}$$

$$a = \underset{a}{\operatorname{argmin}} [\hat{h}(\sigma(n, a)) + c(n_i, \sigma(n, a))]$$

$$\hat{h} = \hat{h}(n) + \sigma(n, a)$$

$$W(n) =$$

$$P(n) = "$$

$$\hat{h}(n) = w_1 W(n) + w_2 P(n) + \dots$$

$$S(n_i)$$

$$n_i$$

$$\hat{h}$$

$$\hat{h}(n_i) \leftarrow \hat{h}(n_i) + \beta \left(\min_{n_j \in \mathcal{S}(n_i)} [\hat{h}(n_i, n_j) + c(n_i, n_j)] - \hat{h}(n_i) \right)$$

$$\hat{h}(n_i) \leftarrow (1 - \beta)\hat{h}(n_i) + \beta \min_{n_j \in \mathcal{S}(n_i)} [\hat{h}(n_j) + c(n_i, n_j)]$$

$$\hat{h}(n_i) \leftarrow \beta \min_{n_j \in \mathcal{S}(n_i)} [\hat{h}(n_j) + c(n_i, n_j)] + (1 - \beta)\hat{h}(n_i)$$

$0 < \beta \leq 1$

)

\hat{h} (

.4

)

(

Rewards

()

()

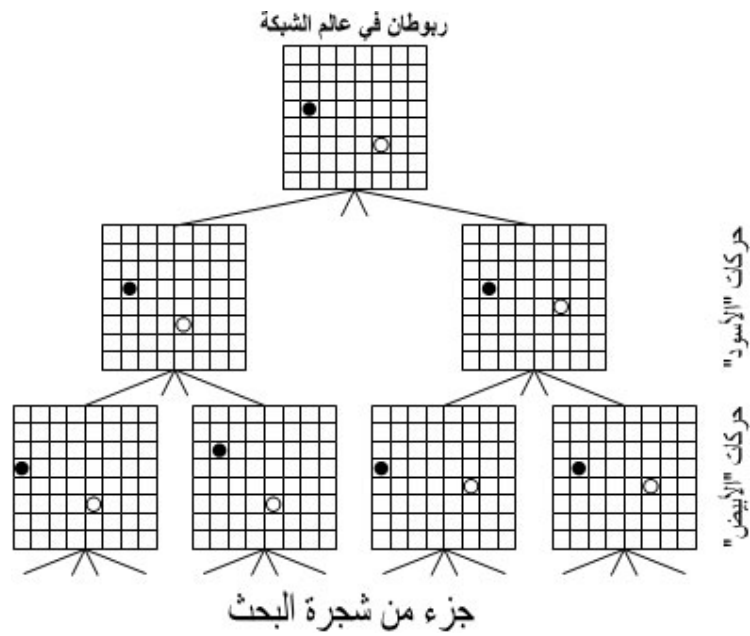
: / /

.()

:Robots

" " " "

()



()

./ /

" " " :

) " .

(

) .

.(

Go

Tic-Tac-Toe

()

min-Max .6

.MAX " "

.MIN MAX

MAX

(MAX) MAX

MIN

.MAX

) .MIN (MIN)

. 2k+1 2k

"k " .(

:

.MIN MAX

()

1022

1040

1/3

.(108)

.()

heuristic

"

"

" "

:

MAX



MIN



.MIN MAX

) .

minimax

.(

MAX

MAX

MIN

MIN

MIN

.()

MAX

MIN

MIN

MAX

MAX

MAX

3x3

(O)

(X)

MIN

(O)

MAX

(X)

MAX

2

:

p

e(p)

:

p



) - (MAX

) = e(p)

(MIN

∞

) $\infty = e(p) : \text{MAX}$

p



(

$-\infty = e(p) : \text{MIN}$

p

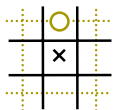
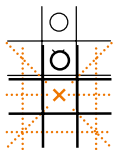


e(p) = 6 - 4 = 2 :

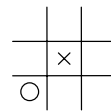
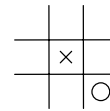
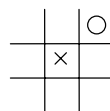
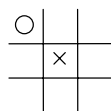
p

MAX

MIN



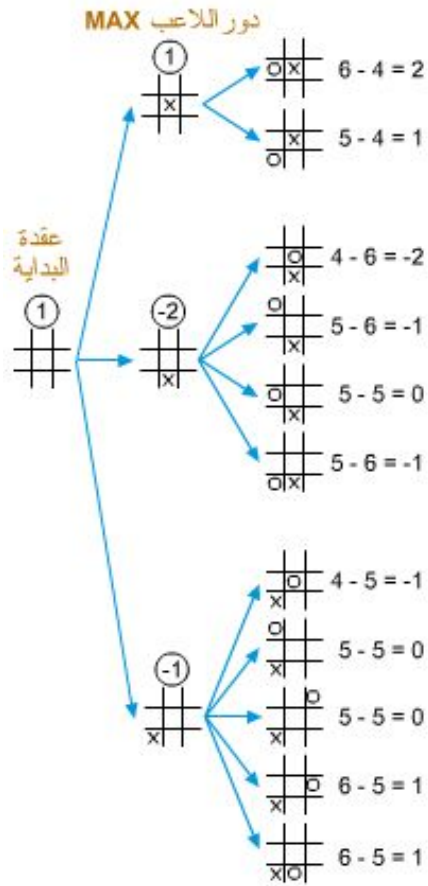
:



.2



MAX



المرحلة الأولى من البحث
في لعبة تيك تاك تو



MIN

2

MIN

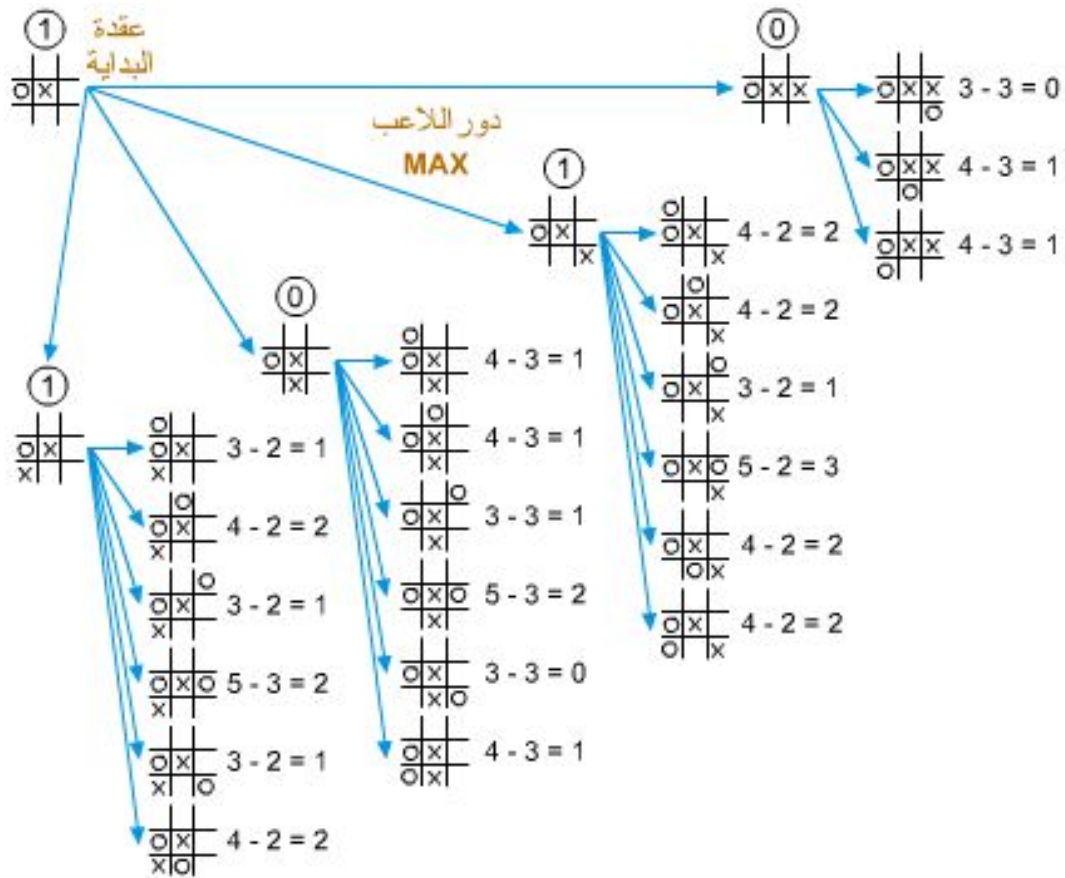
MAX

MAX

" "

MAX

MIN



MIN

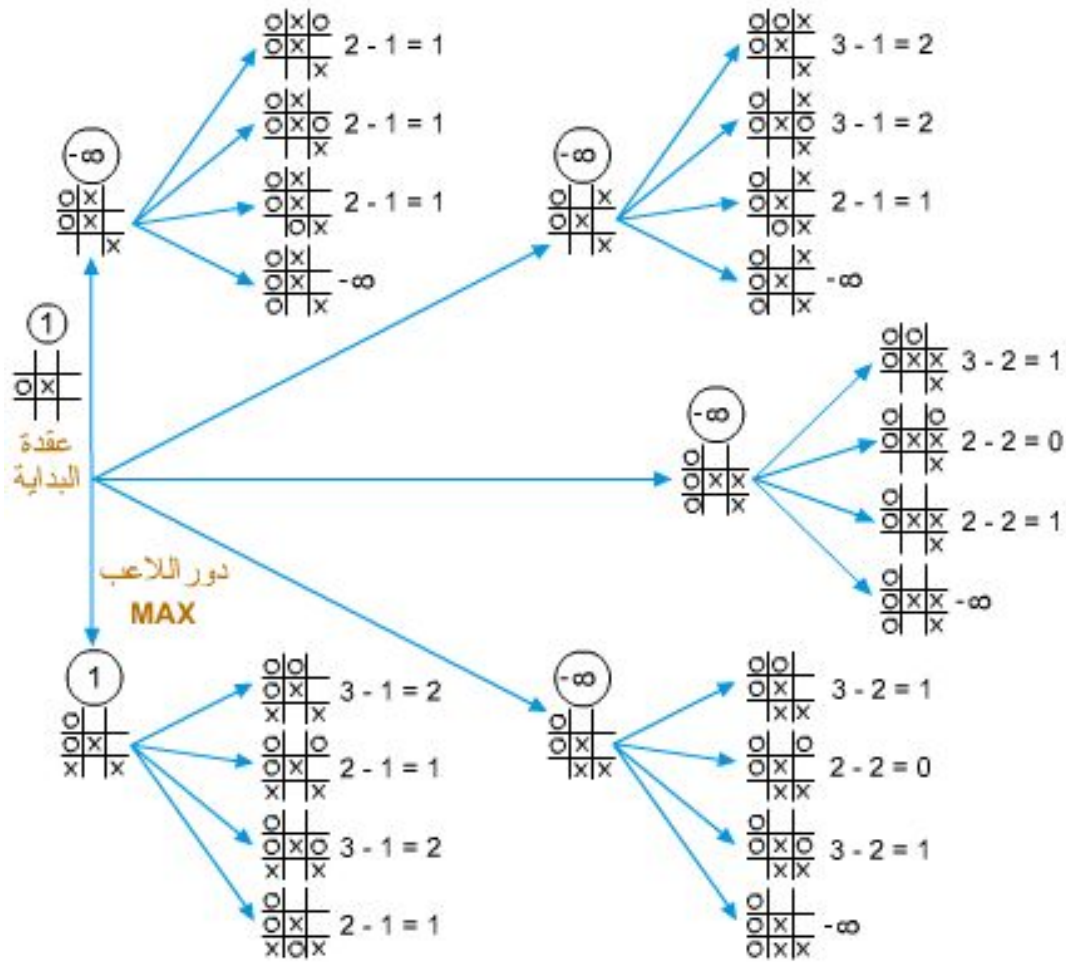
MAX

MIN

MAX

MIN

MAX



المرحلة الأخيرة من البحث
في لعبة تيك تاك تو

.7



: :

.1

:

:



:



FAQ Frequently Asked Questions



:

.2

:

John

Holland

:

.1 0



()



slide : () () : (20)

(slide 21) :

- N .1
- .p_m p_c .2
- x₁, x₂, ..., x_N .N .3
- f(x₁), ..., f(x_N) .4
- .5
- .6
- .7
- .N (5) .8
- .9
- .() (4) .10

255 0

.01111110 h=126 .1

.N=4

h =

0	1	1	1	1	1	1	1	0
---	---	---	---	---	---	---	---	---

 = 126

8 n₀ n₁ 0 f(h) = 8 - |n₁ - n₀| .2

.n₁=n₀=4 8

: h₁, h₂, h₃, h₄ : 4 .3

0	1	1	1	1	1	1	1	0
0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	0
0	0	1	0	0	1	0	0	0

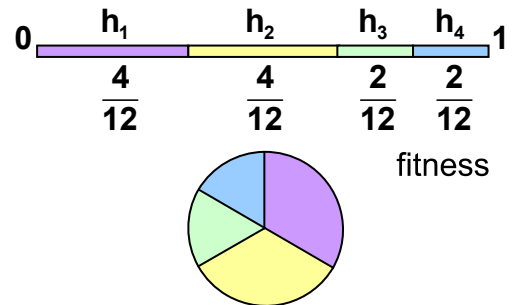
.4, 2, 4, 2: f(h_i) .4

h_1	0	1	1	1	1	1	1	0	$f(h_1) = 8 - 6 - 2 = 4$
h_2	1	1	1	1	1	1	1	0	$f(h_2) = 8 - 7 - 1 = 2$
h_3	0	0	1	0	0	1	0	0	$f(h_3) = 8 - 2 - 6 = 4$
h_4	0	0	0	0	0	0	0	1	$f(h_4) = 8 - 1 - 7 = 2$

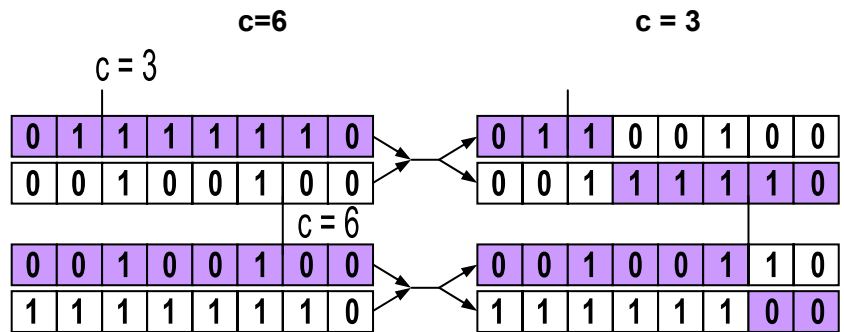
:
$$p(h_i) = \frac{f(h_i)}{\sum_{j=1}^N f(h_j)} \quad .5$$

$p(h_1) = p(h_3) = 4/12$; $p(h_2) = p(h_4) = 2/12$

$r \in [0,1]$ $p(h_i)$ N $[0, 1]$
()



h_1, h_2 h_1, h_3) $.6$



() $.7$

.6, 8, 4, 6 :

0	1	1	0	0	1	0	0	$f = 8 - 3 - 5 = 6$
0	0	1	1	0	1	1	0	$f = 8 - 4 - 4 = 8$
0	0	1	0	0	1	1	0	$f = 8 - 3 - 7 = 4$
1	0	1	1	1	1	0	0	$f = 8 - 5 - 3 = 6$

.8

.3

)

.(

:

LISP

:

5 3

+ list

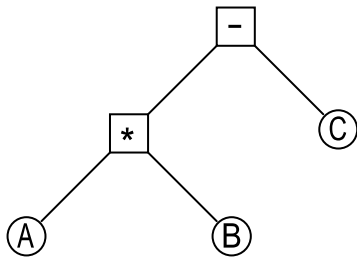
+(3 5)

LISP S-expression (- (* A B) C)

(- (* A :

AB-C

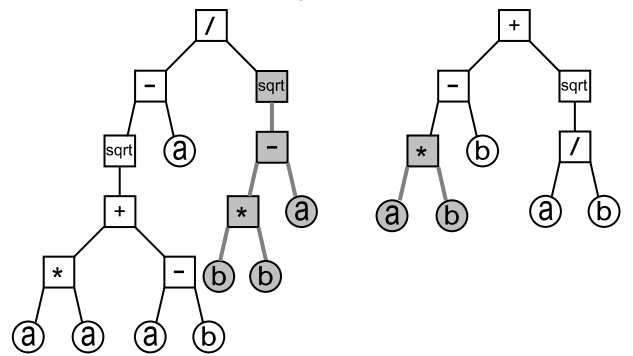
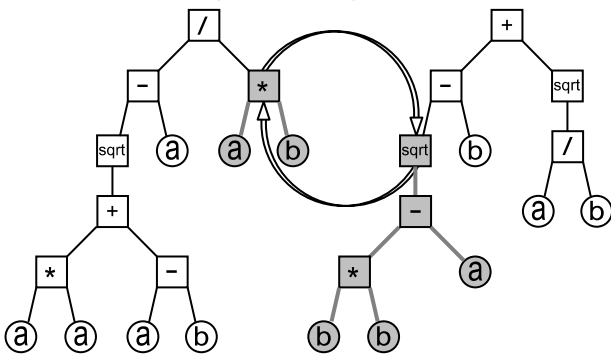
B) C)



:

Two parental S-expressions

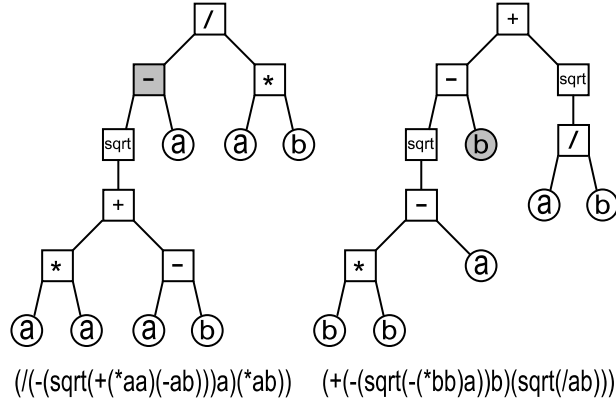
Two offspring S-expressions



((- (sqrt (+ (* aa) (- ab))) a) (* ab)) (+ (- (sqrt (- (* bb) a)) b) (sqrt (/ ab))))

((- (sqrt (+ (* aa) (- ab))) a) (sqrt (- (* bb) a))) (+ (- (* ab) b) (sqrt (/ ab)))

Mutation in generic programming



.4

الخوارزميات الجينية	الشبكات العصبونية	النظم العائمة	النظم الخبيرة	النظم المقارنة	نقاط المقارنة
تحت الوسط	سيء	جيد	فوق الوسط	تمثيل المعرفة	فوق الوسط
جيد	جيد	جيد	فوق الوسط	تحمل معرف غير مؤكدة	فوق الوسط
جيد	جيد	جيد	سيء	تحمل معرف غير دقيقة	سيء
جيد	جيد	تحت الوسط	سيء	تكيف	سيء
جيد		سيء	سيء	إمكان التعلم	سيء

تحت الوسط	سيء	جيد	جيد	إمكان التوسع
فوق الوسط	جيد	تحت الوسط	سيء	كشف المعطيات والتنقيب عنها
فوق الوسط	جيد	فوق الوسط	سيء	الصيانة

.neuro-fuzzy system :



.neuro-expert system :



soft computing

(H)
$ \neg H).p(\neg H)$

.5

.Negnevsky



Michael

artificial Intelligence: a guide to intelligent systems



2002 Addison Wisley

Negnevsky