

Ordinateurs, robots, réseaux informatiques et systèmes mécatroniques, tous ces systèmes non-humains envahissent le domaine de la cognition: calculs, jeux, traductions, vision artificielle, reconnaissance de la parole, capacité à suivre un humain, ou réplication de mouvements enseignés par l'exemple.

Ce domaine de la cognition automatisée, c'est-à-dire de la "cognitive", doit maintenant se baliser de façon précise, quantifiée, technique.

Or qu'est-ce que l'intelligence, l'apprentissage, ou l'abstraction? Comment évaluer la complexité? Combien faut-il de connaissances pour jouer à « pile ou face »? Combien d'informations contient la page que vous avez sous les yeux? Combien en faut-il pour décrire une pipe?

Les théories classiques ne permettent guère de répondre de façon satisfaisante à la plupart de ces questions.

Le présent ouvrage rediscute les bases classiques, puis apporte une contribution nouvelle. Il offre pour la première fois sous forme intégrée, commentée, illustrée de nombreux exemples, et de manière concise, les éléments principaux de la théorie « MCS » pour les sciences cognitives. L'ouvrage fait la synthèse d'éléments progressivement élaborés et démontrés, graduellement publiés par l'auteur dans la littérature scientifique au cours des dernières décennies.

ISBN 978-2-9700629-0-5 Editions Roboptics

Aussi disponible sur [www.lulu.com](http://www.lulu.com)



# La Cognitive

## Définitions et métrique pour les sciences cognitives et la cognition automatisée

Jean-Daniel Dessimoz

```

1: SleepAGN(0.05); break; case
2: if(!SignalIn(NSISstart))
   GoState(6);
   else
   GoState(20); break; case
3: DemarrerMatchAGN(); // start 90 s timer etc.
break; case
21: SignalOutAGN(NSOAspirateur, true) ; // start motor vacuum
break; case
22: SignalOutAGN(NSORouleauIN, true) ; // start motor brush
break; case
23 : ApproAGN(HoleNb1, 15); break; case
24: MoveAGN(HoleNb1); break; case
25: MoveAGN(Trans(173,90,-90)); break; case
26: ObserverLigneAGN(NL, NCStart, NCStop) // visual analysis of a row
if (N2Jaune>0) // totems are yellow; balls are white
{PositionTotemDuBalle[1].TypePosition=Totem;
nbTotem = nbTotem+1;}
else
PositionTotemDuBalle[1].TypePosition=Balle;
break; case

```