



ThanhNT Blog's

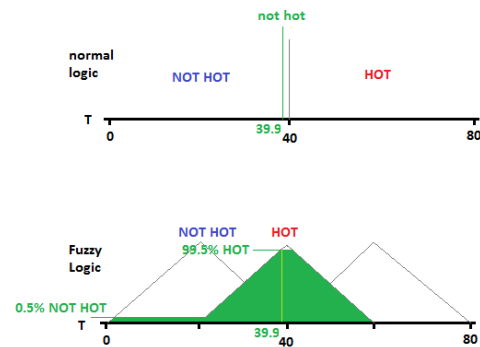
Computer, Networking, Embedded  
Systems, PCB Design



# Cách hiểu đơn giản về Logic Mờ (Fuzzy Logic)

APR 12, 2016 | 19 MIN READ

## Logic mờ (Fuzzy Logic)



Thực sự mà nói, khái niệm này đối với chúng ta (những người không am hiểu sâu về điều khiển tự động) thì quả là cao siêu. (Nghe rất "Mờ" đúng không? :D)

Sau bài viết này tôi nghĩ bạn cũng sẽ đỡ mờ hơn giống như tôi. Bài viết này tôi note lại từ một bài viết của 1 thầy bên Đại học Sư Phạm Kỹ Thuật.

Logic mờ đóng vai trò then chốt trong quá trình điều khiển, hầu hết các kỹ sư về tự động hoá hay cơ khí, cơ điện tử đều nắm rõ khái niệm này và có thể đưa ra một quy luật điều khiển phù hợp.

## Logic mờ là gì hở con



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



Nhà tôi vừa sắm một chiếc máy giặt để cho mẹ “nhàn hơn”. Quyển hướng dẫn bảo “Máy này hoàn toàn tự động, hoạt động thông minh với logic mờ, bạn chỉ cần bỏ quần áo vào, ấn một nút và... đi ngủ rồi thức dậy lấy quần áo sạch ra phơi. Máy tự tính ra chế độ giặt và lo liệu tất cả”.

Mẹ tôi hỏi “Lo...gíc mờ là cái gì hở con?”. “Dạ, logic mờ là ...”, tôi gãi đầu, “thì là logic mờ!”. Mẹ tôi cau mày “Mày cũng chẳng hơn gì tao, đúng là đồ... gà mờ!”.

Tôi phải giải thích thế nào nhỉ? Có thể những thứ viết dưới đây thật ngô nghê với những kỹ sư tin học hay tự động hoá, nhưng là để cho những người như mẹ tôi “cảm nhận” thế nào là logic mờ, thứ mà ở đâu cũng nghe người ta quảng cáo là làm cho máy giặt, máy điều hoà, máy sưởi, lò vi sóng ... của họ “thông minh” hơn, và tất nhiên, giá cũng mắc hơn !

Bây giờ con kể mẹ nghe một chuyện cổ tích ...(hm, ngày xưa thì mẹ kể con nghe)

Ngày xưa ngày xưa, có một cô gái trẻ đi lang thang khắp thế giới. Cô không có một người thân hay thứ tài sản gì, khi tới một xứ sở u ám - nơi hoàng tử đang bị một con rồng giam giữ, cô quyết định thử vận may đi cứu chàng. Dù đã vô vọng, nhà vua hứa gả hoàng tử và chia nửa giang sơn cho bất kỳ ai cứu được người con trai duy nhất của mình.



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



Cô gái tới chỗ con rồng xin nó thả hoàng tử ra. Con rồng thương hại nhìn cô gái và nói : “Trước hết cô phải giải một câu đố của ta. Nếu cô giải được, hoàng tử sẽ thuộc về cô. Còn không thì ta sẽ ăn thịt ngươi. Nếu ta nói rằng tất cả rồng ở xứ này đều nói dối, vậy ta đúng hay sai? “. Cô gái trả lời “Thật đơn giản. Nếu ngươi dùng logic Arít-xtốt thì ngươi sẽ mâu thuẫn mãi mãi, nhưng làm mờ nó đi một chút ngươi sẽ thấy ở một mức độ nào đó ngươi đang nói dối, và cũng đang nói thật ở một mức độ khác”. Con rồng hài lòng với câu trả lời khôn ngoan của cô gái. Nó bèn thả hoàng tử ra rồi cùng về lâu đài dự lễ kết hôn của hai người.

Ô, không phải rồi, hay là chuyện một chàng trai đi cứu nàng công chúa chứ? Tôi không rõ nữa, nhưng câu chuyện vẫn kết thúc với đám cưới của đôi trai gái và không khí hoan hỉ khắp xứ sở.

## Logic nhị phân

Cô gái nói “logic Arít-xtốt” và “làm mờ đi” nghĩa là thế nào nhỉ? Arít-xtốt là một học giả Hy-lạp được coi là cha đẻ của logic, hay chính xác hơn là logic nhị phân. (À quên, chưa nói logic là gì. Logic là cách lập luận để ai cũng đồng tình với mình, đúng không nhỉ?) Logic nhị phân dựa trên một luật rằng mọi thứ hoặc là A hoặc là không-A. Đó là thứ logic bạn phải chấp nhận khi làm phép thử đúng-sai. Đó cũng là logic của các máy tính: có-điện hoặc không-có-điện, 1 hoặc 0. Tới



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



giờ, logic nhị phân vẫn rất đúng với nhiều thứ. Thí dụ, nếu bạn hỏi trong lớp “Ai là con gái?”, tất cả em gái sẽ giơ tay lên và các em trai bỏ tay xuống. Bạn có một câu trả lời dứt khoát, bởi mỗi người hoặc là con gái hoặc không là con gái. Thế nếu bạn hỏi “Ai thích đi học?” Một vài đứa có thể giơ cả hai tay (chúng rất thích đi học) và những đứa khác thì có thể hạ tay xuống (chúng ghét đi học). Tuy vậy hầu hết bọn trẻ sẽ giơ tay lên, rồi hạ xuống vài lần và sau đó để lơ lửng ở giữa. Nói chung thì chúng nó thích học, nhưng lại có một số thứ ở trường làm chúng ghét.

Nếu muốn ghi lại kết quả theo logic nhị phân thì bạn phải bắt mỗi đứa hoặc là thích học hoặc là ghét học; A hoặc không-A mà. Nhưng ở đây cần phải có một thứ logic khác để ghi chính xác câu trả lời; logic này cho phép lũ trẻ có thể vừa thích học vừa đồng thời không thích học. Đó là ...

## Logic mờ

Logic mờ dựa trên lập luận rằng A có thể chứa không-A. Nghĩa là một thứ có thể chứa một phần thứ khác mâu thuẫn với nó. Như khi cô gái trong câu chuyện nói nếu giải câu đố theo logic mờ, thì con rồng vừa nói dối (A) vừa không nói dối (không-A).

Chúng ta cùng xem lại tình huống ở lớp học, lần này với logic mờ. Ta đã biết rằng có thể phát biểu một đứa trẻ cùng một lúc vừa thích đi học vừa không thích đi học. Tuy vậy



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



vẫn chưa đủ chính xác, vì một đứa có thể thích nhiều hơn là ghét, còn đứa khác thì ngược lại. Để giải quyết chuyện này, ta dùng các chân lý mờ. Chân lý mờ là một cách để biểu thị mức độ một thứ có A và mức độ nó có không-A. Thường thì ta ghi chân lý mờ bằng ký hiệu phần trăm. Do vậy, bây giờ ta có thể nói là đứa này thích đi học 75% và ghét đi học 25%, còn đứa kia thì thích 45% và ghét 55%. Tổng các chân lý mờ (yêu + ghét) phải bằng 100%. Tất nhiên, không ý nghĩa gì lắm khi dùng số để nói ai thích đi học tới mức nào, do nó không thật rõ ràng, nhưng còn hơn là cực đoan: chỉ có thích đi học hoặc không thích đi học.

Một cách khác để mô tả thực tế khách quan chính xác hơn là dùng quan hệ thành viên của các tập mờ. Một tập nhị phân và một tập mờ khác nhau ở chỗ: trong một tập "chuẩn" mỗi phần tử hoặc là thuộc hoặc là không thuộc tập đó. Ở đây ta lại gặp A hoặc là không-A. Trong một tập mờ, một phần tử có thể là thành viên của tập ở mức độ nào đó và đồng thời lại không là thành viên của tập ở một mức độ khác.

Nếu muốn minh họa một tập người lớn bằng tập nhị phân, ta có một đồ thị xung vuông. Nghĩa là lớn hơn hoặc bằng 18 tuổi đều là người lớn và nhỏ hơn là trẻ con. Trong đó giả thiết một người là "lớn" khi hơn 18 tuổi. Dĩ nhiên ai đó có thể tranh cãi là một người thành người lớn khi 21, thì lúc đó ta lại phải thay đổi đồ thị. Nhưng có một điều giữ



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



nguyên, là mỗi người hoặc là người lớn hoặc không phải người lớn, biểu diễn trong đồ thị là 1 hoặc 0.

Khi ta vẽ đồ thị tập mờ các người lớn, ta có một đồ thị hình thang hoặc tam giác tùy theo quy luật phân bố. Ở đây có một quá trình trung gian giữa là người lớn và không phải người lớn. Lại lần nữa, ta có thể tranh cãi là phải vẽ đoạn cong thế nào cho chính xác. Ai đó có thể nói 13 tuổi hoàn toàn không phải người lớn hoặc 19 tuổi thì phải coi là người lớn rồi. Nhưng ta có thể chắc chắn rằng đường cong mờ của tập người lớn gần với chân lý hơn là đường cong nhị phân; tất cả chúng ta đều đồng ý là không thể đưa ra ngày giờ chính xác khi một người trở thành người lớn. Tất nhiên không thể lúc đi ngủ còn là trẻ con và hôm sau tỉnh dậy đã là người lớn. Trưởng thành là một quá trình từ từ và các tập mờ mô tả các quá trình từ từ này chính xác hơn.

Ở trên, ta mới chỉ xét một khía cạnh về tuổi: ở tuổi nào thì bạn là người lớn và ở tuổi nào thì bạn không phải người lớn. Dĩ nhiên, có nhiều khía cạnh về tuổi tác: già hoặc không già, trẻ con hoặc không trẻ con, khi nào thì bắt đầu trung niên ... Tất cả những thứ đó là thí dụ về những tập con khác nhau của tập tuổi tác.

Ta có thể vẽ đồ thị các tập con khác nhau như tập con "Trẻ con" và "Người lớn" trên cùng một đồ thị. Về mặt lý thuyết, ta có thể nhét toàn bộ các tập con của tập "Tuổi tác"



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



vào cùng 1 đồ thị. Tất nhiên thế thì rối rắm và ta khó mà hiểu được đồ thị, do vậy thường giới hạn một số lượng tập con nhất định trên mỗi đồ thị.

## Chương trình máy tính và logic mờ

Để bạn hiểu logic mờ được dùng như thế nào trong các chương trình máy tính, chúng ta xét thí dụ trò chơi chó Chicalm. Trước hết ta xem một số thực tế về hành vi của chó. Khi hai con chó gặp nhau, chúng dùng nhiều cử chỉ để cho con kia biết mình cảm thấy thế nào. Các tín hiệu quan trọng nhất cho biết về sự bình thản, nghĩa là con chó kia không có gì nguy hiểm. Trong trò chơi này, bạn phải điều khiển chú cún Chihuahua. Bạn có thể bắt nó nằm xuống, ngồi xuống, đi tới chỗ chơi hoặc lúc lắc đầu (tất cả đều là những tín hiệu xoa dịu). Khi gặp con chó to, bạn phải làm cho nó ứng xử sao cho không bị con to ăn thịt.

Chương trình điều khiển con chó to xem có ăn thịt Chihuahua bé nhỏ hay không. Để làm thế, chương trình theo dõi mức độ tức giận của chó to. Sau mỗi ván, mức độ tức giận ban đầu lại cao hơn một chút. Mức độ bình thản luôn tính bằng cách trừ đi mức độ tức giận. Nếu mức độ tức giận quá cao thì ... bye, bye Chihuahua tội nghiệp!

Để tính mức độ bình thản, chương trình dùng logic mờ. Chó phản ứng khác nhau trước các





## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



tín hiệu xoa dịu tùy theo khoảng cách giữa chúng. Ở khoảng cách gần, chúng phản ứng tốt với những cái lắc đầu, trong khi ở khoảng cách xa nằm xuống là tốt nhất. Ở khoảng cách trung bình, hãy tới chỗ chơi và ngồi xuống là tốt nhất. Trò chơi Chicalm dùng đồ thị dưới đây để quyết định mức bình thần của một tín hiệu xoa dịu.

Khi chó tức giận, chúng không nghĩ rõ ràng như vậy. Nghĩa là khi chó to nổi cáu, thậm chí ở khoảng cách nhỏ, thì chó con cũng nên nằm xuống; lúc đó phải có tín hiệu xoa dịu mạnh hơn. Để đưa hành vi này vào trò chơi, mức độ giận dữ của chó to được thêm vào khoảng cách khi tính mức xoa dịu.

Tín hiệu cuối cùng mà chó to có thể chấp nhận là bỏ chạy. Giữa bọn chó, việc bỏ chạy trước khi hít hít mũi là rất thô lỗ! Với mỗi bước Chihuahua làm, con chó to theo dõi và mức giận dữ của nó tăng lên nếu Chihuahua không xoa dịu “đủ độ”, và khi đó chú không có cơ hội để chuồn qua. Dĩ nhiên, sau khi chó to hít hít xong, nó sẽ không để ý gì tới Chihuahua nữa. Lúc này Chihuahua có thể chạy an toàn tới phía kia của màn hình.

## Logic mờ trong máy giặt là thế này, các bạn à

Ngày nay nhiều thiết bị gia dụng có logic mờ để dễ dùng hơn. Bạn có thể tìm thấy logic mờ trong đầu vòi hoa sen, nồi cơm điện, máy hút bụi và nhiều thứ máy thông minh khác.





## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



Để biết những máy này làm việc như thế nào, chúng ta sẽ xem một mô hình đơn giản hoá về máy giặt logic mờ.

Giống như máy giặt mờ thực sự, mô hình của chúng ta trước hết kiểm tra xem đồ giặt bẩn đến đâu. Một khi biết được độ bẩn, nó sẽ dễ dàng tính ra thời gian cần để giặt.

Đầu tiên nó luôn lấy thời gian tối thiểu là 10 phút. Làm thế để người dùng luôn hài lòng với công việc của nó, dù khi họ bỏ đồ sạch nguyên vào. Sau đó nó tính mức độ bẩn. Nếu bẩn 100%, nó cộng 2 phút cho mỗi đồ giặt. Tất nhiên, một máy giặt thực sự sẽ thực hiện những tính toán này cuối cùng, nhưng mô hình của chúng ta thực hiện cho từng đồ giặt để bạn tiện theo dõi.

Nếu bạn cho một đồ bẩn 50% thì nó sẽ thêm 50% của 2 phút vào: tức là cộng 1 phút vào 10 phút. Máy giặt của chúng ta không chỉ kiểm tra bụi bẩn mà cả dầu mỡ. Đồ giặt dính dầu mỡ sẽ phải giặt lâu hơn. Do đồ giặt có thể vừa dính bụi bẩn vừa dính dầu mỡ, ta cho cả hai thứ vào một đồ thị:

Bạn thấy ở đồ thị bên phải: điểm (0, 0) là đồ giặt hoàn toàn sạch sẽ, không bụi bẩn, không dầu mỡ; điểm (0, 1) là khi đồ giặt không dính dầu mỡ nhưng bẩn; điểm (1, 0) — dính dầu mỡ nhưng không bẩn; và điểm (1, 1) - vừa bẩn vừa dính dầu mỡ. Máy giặt cộng 2 phút cho mỗi đồ giặt nếu bẩn 100% hoặc dính mỡ 100%, và cộng 4 phút nếu bẩn và



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



dính dầu mỡ 100%. Bây giờ, nếu bạn có đồ  
bẩn 100% và dính dầu mỡ 50% thì máy cộng  
thêm  $2 \times 100\% + 1 \times 50\% = 3$  phút vào thời gian  
tối thiểu 10 phút.

Chiếc máy giặt lý thuyết của chúng ta chỉ xét  
hai thứ trên. Với máy giặt thật, nó còn kiểm  
tra xem cần bao nhiêu xà phòng, cần thêm  
bao nhiêu nước, phải quay nhanh thế nào,  
theo hướng nào ... Nếu muốn vẽ đồ thị cho  
từng tính toán của máy giặt logic mờ thì ta  
sẽ có một siêu hộp khá phức tạp. Vì thế ở  
đây ta chỉ giới hạn ở hai yếu tố: bụi bẩn và  
dầu mỡ.

Trong hai thập kỷ qua, máy giặt logic mờ  
phát triển bùng nổ. Không chỉ do khả năng  
thay thế con người về cả mặt tính toán và  
thao tác khi giặt, mà chúng còn rẻ hơn nhiều  
so với những máy giặt nhị phân thông  
thường.

## Lịch sử của logic mờ

### Ai nói tới logic mờ đầu tiên

Ngày nay khi nhìn lại lịch sử của logic mờ,  
người ta nhận thấy người đầu tiên đề cập tới  
logic mờ chính là Đức Phật (500 năm trước  
CN). Triết lý Phật giáo dựa trên tư tưởng rằng  
thế giới đầy những mâu thuẫn, “sắc không  
không sắc”, mọi thứ đều chứa một phần đối  
lập của nó. Bước chân vào mỗi ngôi chùa  
chúng ta đều thấy ở ngay gian trước là hai vị  
Thiện — Ác, là hình ảnh hai mặt tốt và xấu  
trong mỗi con người.



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



Nói theo lý thuyết logic mờ nghĩa là sự vật có thể đồng thời là A và không-A. Ở đây ta thấy có một mối liên hệ rõ ràng giữa triết lý Phật giáo và logic mờ hiện đại. Thuyết âm dương của người Trung Quốc cũng hàm chứa logic mờ!

“Logo” bát quái thể hiện tư tưởng cốt yếu của thuyết:

- hình tròn thể hiện sự toàn vẹn của sự vật, trời đất; mỗi sự vật hiện tượng đều có hai mặt âm và dương đối lập nhau, cùng tồn tại, mặt này thịnh thì mặt kia suy (phần âm to ra thì phần dương nhỏ đi và ngược lại).
- dấu trắng trong phần đen và dấu đen trong phần trắng thể hiện trong âm có dương, trong dương có âm.
- dấu đen trong đầu to của phần trắng thể hiện khi dương cực thịnh thì chính là lúc trong lòng nó xuất hiện âm (và ngược lại).

Sau đức Phật 200 năm, nhà bác học Hy-lạp là Arít-xtốt phát triển logic nhị phân. Trái ngược với triết lý nhà Phật, Arít-xtốt cho rằng thế giới tạo bởi các đối nghịch, thí dụ nam-nữ, nóng-lạnh, khô-ướt. Mọi thứ hoặc là A hoặc là không-A, không thể cả hai.

Logic nhị phân của Arít-xtốt trở thành nền tảng cho khoa học, nếu một thứ được chứng minh về mặt logic (nhị phân) thì nó được và vẫn sẽ được khoa học công nhận. Cho tới



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



cuối thế kỷ 19, khi một nhà văn-nhà toán học người Anh, Russel, phát hiện ra một nghịch lý của logic nhị phân...

### Russel (1872-1970), người khai sinh logic mờ

Bá tước Bertrand Arthur William Russel sinh ra trong một gia đình quý tộc Anh năm 1872. Ông có một cuộc đời dài và đầy biến động. Thời trẻ tuổi, ông nghiên cứu toán học và sau đó, cùng với một nhà toán học khác, viết một cuốn sách về những cơ sở của toán học. Trong sách, họ dành cả một trang chỉ để chứng minh  $1 + 1 = 2$ . Trong quá trình nghiên cứu, ông đã phát hiện ra một nghịch lý mà ngày nay gọi là nghịch lý tập của Russell.

Trước hết chúng ta phân biệt hai loại tập: tập chứa chính nó và tập không chứa chính nó.

**Xét thí dụ:** một quả lê thuộc tập các quả lê, nhưng tập các quả lê không thuộc về tập các quả lê do bản thân nó không phải là một quả lê! Nghĩa là tập các quả lê không phải là một thành viên của chính nó.

Bây giờ ta xét một tập khác, tập mọi thứ không phải quả lê, gồm sách, chuột cống, hay cả tổng thống Bush nữa! Do trong tập này bạn tìm thấy mọi thứ không phải quả lê, nên bạn cũng có thể tìm thấy trong đó tập các quả lê và tập mọi thứ không phải quả lê! Nghĩa là tập mọi thứ không phải quả lê là thành viên của chính nó.



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



Russel đi sâu hơn và xem xét tập của mọi tập mà không chứa chính nó. Trong tập này, bạn sẽ tìm thấy tập các quả lê, tập các tổng thống, và nhiều tập khác nữa. Nhưng bạn sẽ không tìm thấy tập mọi thứ không phải quả lê, do tập đó chứa chính nó và do vậy không thoả mãn tiêu chuẩn đặt ra.

Trong khi xem xét tập các tập không chứa chính nó này, Russell băn khoăn liệu nó có phải là một thành viên của chính nó?

Nếu nó là một thành viên của chính nó, thì không thoả mãn định nghĩa. Mặt khác, nếu nó không phải là thành viên của chính nó, thì theo định nghĩa về tập đó, thì nó lại thoả mãn và như vậy nó là thành viên của chính nó!

Vì vậy khi tìm ra nghịch lý này, Russell ngẫu nhiên chứng minh rằng logic nhị phân, mà ông nghĩ là cơ sở của toán học, không thể tự chứng minh nó.

Tất nhiên ngày nay, chúng ta biết nghịch lý của Russell không phải là một trường hợp không giải được, nếu dùng logic mờ thì ta có câu trả lời ngay. Tuy nhiên, Russell không hề biết gì về logic mờ và đã vô cùng thất vọng với toán học. Ông từ bỏ toán học, nhưng như thế không có nghĩa là ông đã dừng lại việc làm đảo lộn thế giới này. Trong suốt cuộc đời 97 năm, ông luôn truyền bá tư tưởng của mình; ông viết hàng tá sách, sách toán, triết luận, tiểu thuyết, thậm chí cả thứ sách lá cải nữa. Khi mất năm 1970, ông đã không chỉ



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



khởi đầu một trang mới của logic học, mà còn đoạt cả một giải Nobel văn học. Ông là một thí dụ điển hình cho thấy những người có tài năng lớn về toán học cũng có thể là những nhà văn lớn.

### Zadeh, cha đẻ của logic mờ hiện đại

Năm 1964, giáo sư Zadeh bắt đầu suy nghĩ liệu có thứ logic tốt hơn nào dùng trong máy móc. Ông có ý tưởng liệu ta có thể bảo máy điều hoà làm việc nhanh hơn khi trời nóng lên, hay những vấn đề tương tự như thế, sẽ hiệu quả hơn việc đặt ra từng luật cho từng nhiệt độ. Đây chính là bước đi đầu tiên của logic mờ hiện đại như chúng ta hiểu và ứng dụng ngày nay.

Phải mất một thời gian dài logic mờ mới được chấp nhận, mặc dù ngay từ đầu một số người đã rất quan tâm. Bên cạnh các kỹ sư, những nhà triết học, tâm lý học và xã hội học nhanh chóng áp dụng logic mờ vào ngành khoa học của mình.

Năm 1987, Nhật Bản đã xây dựng hệ thống tàu điện ngầm đầu tiên làm việc với hệ thống điều khiển hoạt động tàu tự động dựa trên logic mờ. Đây là một thành công lớn và dẫn tới sự phát triển bùng nổ của logic mờ. Các trường đại học và các hãng công nghiệp đua nhau phát triển những ý tưởng mới.

Đầu tiên là ở Nhật Bản, do tôn giáo ở Nhật thừa nhận rằng mọi thứ có thể chứa phần đối lập của chính nó, chứ không coi là một



## ThanhNT Blog's

Computer, Networking, Embedded  
Systems, PCB Design



thứ “kinh khủng” như hầu hết những nơi khác trên thế giới. Và logic mờ cũng hứa hẹn đem lại nhiều tiền bạc cho các hãng công nghiệp, tất nhiên là điều này được đón chào.

### Thay lời kết

Ngày nay, hầu hết máy móc thông minh đều chứa công nghệ logic mờ. Nhưng logic mờ không chỉ giúp nâng cao khả năng suy diễn của máy móc. Nếu chúng ta có thể từ bỏ ý nghĩ là mọi thứ phải hoặc là tốt hoặc là xấu, ta sẽ nhìn thấy những điều tốt trong mọi người.

Chúng ta sẽ không còn phải bắt mọi người hoặc phải là thiên thần hoặc phải là quỷ dữ. Mỗi người có những điểm tốt và điểm xấu của riêng mình.

Và việc của chúng ta là phát hiện ra chúng! Đó là ứng dụng đẹp nhất của logic mờ trong cuộc sống.

Từ [invisionfree.com](https://invisionfree.com)

[Fuzzy Logic](#)

[Logic Mờ](#)

