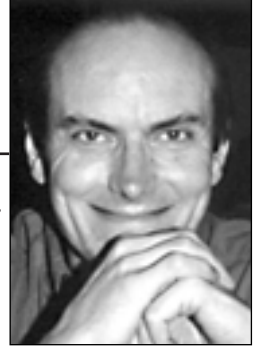


Bilgisayar Bilimi Köşesi

Chris Stephenson ve Ali Nesin*
cs@cs.bilgi.edu.tr, anesin@bilgi.edu.tr



Köşeyi Dönen Robot

Ağustos 2002'de kaybettiğimiz büyük bilgisayar bilimcisi Edgar W. Dijkstra, "Astronominin teleskoplarla ilişkisi ne kadarsa, bilgisayar biliminin de bilgisayarlarla ilişkisi o kadardır!" demişti.

Bilgisayar bilimi bilgisayarlarla ilgili değilse ya neyle ilgilidir? Özünde, bilgisayar bilimi iş yapma yöntemleri hakkındadır. Bu işlerin, her söylenene itaat eden bir insana ya da bir makineye emir (komut) vererek yapıldığı düşünülebilir. Sorduğumuz sorular bunlar: İş nasıl tanımlanır? Bir iş en güzel ve açık şekilde nasıl ifade edilebilir? Bu işi yapmak mümkün müdür? Bir işi gerçekleştirmek için verilen komutlar nasıl ifade edilmeli ve makina tarafından nasıl kavranmalı? Verilen komutlar her koşulda bu işi gerçekleştirir mi? Gerçekleştirirse bunu kanıtlayabilir miyiz? Bu işi en ekonomik biçimde ve en az zamanda nasıl gerçekleştirebiliriz?

Şu son derece basit örnek üzerinde düşünelim:

Bir robot, hem sağına hem soluna doğru sonsuz uzanan bir duvarın önünde. Duvarda her metrede bir bir kapı var. Kapılardan sadece biri açık, diğerleri kapalı. Açık kapı ya sağda ya solda, az ya da çok ileride, ama kesinlikle kapılardan biri açık, nerde olduğunu bilmiyoruz. Robot açık kapıyı bulmalı, görevi o. Ve ne yazık ki robot kapının açık olduğunu

kapının ancak tam yanındayken anlayabiliyor. Robotu nasıl programlamalıyız ki robot en az mesafe yürüyerek açık kapıyı bulsun?

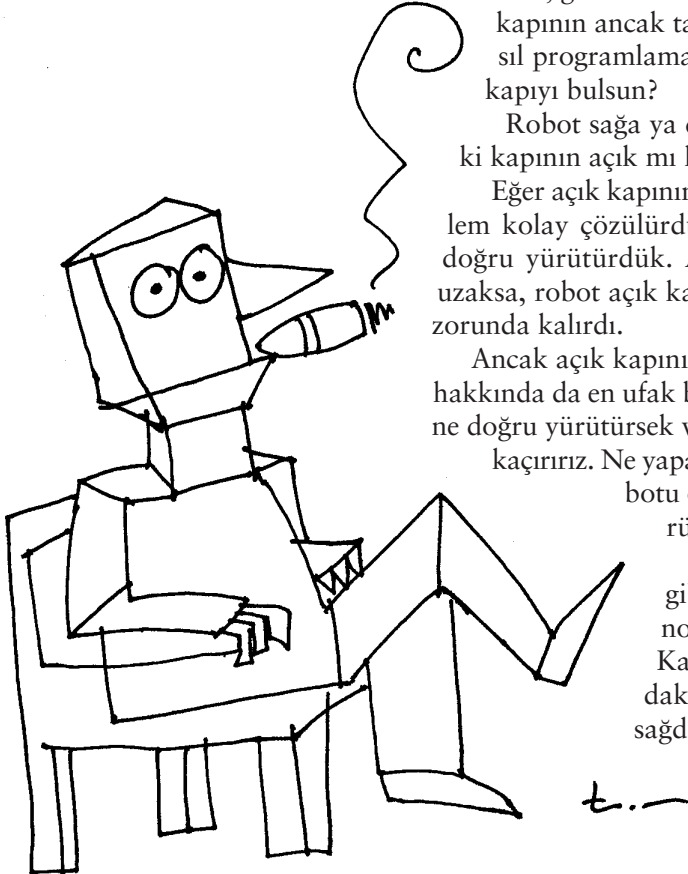
Robot sağa ya da sola gidebilir. Her gittiği yerde önündeki kapının açık mı kapalı mı olduğunu anlayabilir.

Eğer açık kapının kesinlikle solda olduğunu bilseydik, problem kolay çözüldü. Robotu açık kapıyı bulana kadar sola doğru yürütürdük. Açık kapı başlangıçta robottan ne kadar uzaksa, robot açık kapıyı bulmak için o kadar mesafe yürümek zorunda kalırdı.

Ancak açık kapının ne tarafta olduğunu bilmiyoruz, uzaklığı hakkında da en ufak bir fikrimiz yok. Eğer robotu sadece tek yöne doğru yürütürsek ve açık kapı "yanlış" taraftaysa açık kapıyı kaçıırız. Ne yapabiliriz? Belli ki açık kapıyı bulana kadar robotu önce sola, sonra sağa, sonra tekrar sola yürütmemiz gerekli.

Şu stratejiyi irdeleyelim: Duvarı bir doğru gibi görelim. Robotun bulunduğu noktaya 0 noktası diyelim. Her metrede bir bir kapı var. Kapıları tam sayılarla numaralandıralım. Soldaki kapıların numaraları eksi sayılar olsun, sağdakilerin artı. Robot, önce 1 kapısına gitsin

ve o kapı açık mı diye baksın (robot 1 metre yürüdü.) Kapalıysa, robot -1 kapı



* İstanbul Bilgi Üniversitesi öğretim üyesi.

sına gitsin ve o kapıya baksın (robot 2 metre daha yürüdü). Kapı gene kapalıysa, bu sefer öbür istikamete gitsin, ta 2 kapısına kadar (robot 3 metre daha yürüdü). Kapıyı açık bulursa ne âlâ. Bulamazsa -2 kapısına gitsin (robot 4 metre daha yürüdü). Ve robot böylece devam etsin. Açık kapıyı bulmak için en fazla kaç metre yürüyecek?

Yürünen Mesafe	Kapı
1	1
2	-1
3	2
4	-2
5	3
6	-3
7	4
8	-4

Diyelim ki açık kapının numarası $-n < 0$. Açık kapının numarası pozitif olsaydı, ufak bir fark olsa da, açık kapı daha kısa sürede bulunacaktı. Fakat biz en uzun mesafeyle ilgilen-

yoruz.

Robot, yukardaki yöntemle, $1 + 2 + 3 + \dots + 2n$ metre yürüdükten sonra $-n$ numaralı açık kapıya ulaşacaktır, yani toplam

$$\frac{2n(2n+1)}{2} = n(2n+1) = 2n^2 + n$$

metre sonra.

Pek iyi bir sonuç sayılmaz. Eğer -1000 numaralı kapı açıksa, ki bu kapı robotun başlangıç noktasından sadece 1 kilometre uzakta, robotun açık kapıyı bulması için bu yöntemle 2.001.000 metre yani 2000 kilometreden fazla yürümesi gerekecek.

Üstelik açık kapı ne kadar uzak olursa, bu stratejimiz o kadar kötü sonuç verecektir. Çünkü

yürünen uzaklık için bulduğumuz formül, açık kapının başlangıç noktasına olan uzaklığının karesi kadar büyüyor. Açık kapının numarası ikiye katlandığında, robotun yürümesi gereken mesafe dörde katlanacak. Açık kapıyı bulmanın daha kısa bir yolu var mı?

Problemi daha açık sunalım. Örneğimizde olduğu gibi, başlangıç noktasıyla açık kapı arasındaki uzaklığın karesine bağımlı olan bir arama yöntemi yerine, doğrusal bağımlı olacak bir arama yöntemine “daha iyi bir yöntem” diyelim. Daha açık bir ifadeyle: Eğer açık kapı, robotun bulunduğu yerden n metre uzaktaysa, öyle bir strateji (algoritma) istiyoruz ki, (1) robot açık kapıyı bulsun ve (2) robotun bu stratejiyi izleyerek n mesafe ilerdeki açık kapıyı bulmak için gideceği $f(n)$ mesafesi, bir C sabiti için.

$$f(n) \leq Cn$$

koşulunu sağlasın.

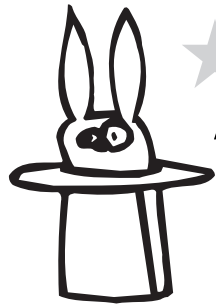
C sabitinin n 'den bağımsız olması gerektiğine dikkatinizi çekerim.

Bu “daha iyi” tanımıyla daha iyi bir yöntem geliştirebilir misiniz? C 'nin değeri nedir? C 'yi ne kadar küçültebilirsiniz?

Yanıtlarınızı, yorumlarınızı, düşüncelerinizi ve sorularınızı cs@cs.bilgi.edu.tr e-posta adresine yollayın lütfen.

Ayrıca, bu köşede görmek istediklerinizle ilgili düşüncelerinizi bekliyoruz. Programlama örnekleri görmek ister misiniz? İsterseniz, hangi dili kullanmalıyız?

Bizim “en iyi” olarak adlandırdığımız çözümü bir sonraki sayıda vereceğiz. Belki sizinki bizimkinden daha iyi olur, kimbilir! ♠



Abra Kadabra

Murat Kipel*
mkipel@tradeorbis.com

52 kartlık standart bir iskambil desteniz var. Bu kartlarla asistanınızın da yardımıyla seyircilerinize küçük bir numara yapacaksınız.

Seyircilerden biri iyice karışmış deste içinden rastgele beş kartı asistanınıza verecek. Asistan da bu beş karttan birini seçerek cebine koyacak ve diğer 4 kartı istediği sırada tek tek size gösterecek. Siz de cebindeki son kartın ne olduğunu şıp diye söyleyivereceksiniz. Asistanla aranızda kartları göstereceği sıra dışında önceden belirlenmiş bir şifre (mimikler, el hareketleri vb.) olmamalı.

Bu numarayı yapmak için nasıl bir yöntem izlemelisiniz? ♠

* TradeOrbis, yazılım uzmanı.