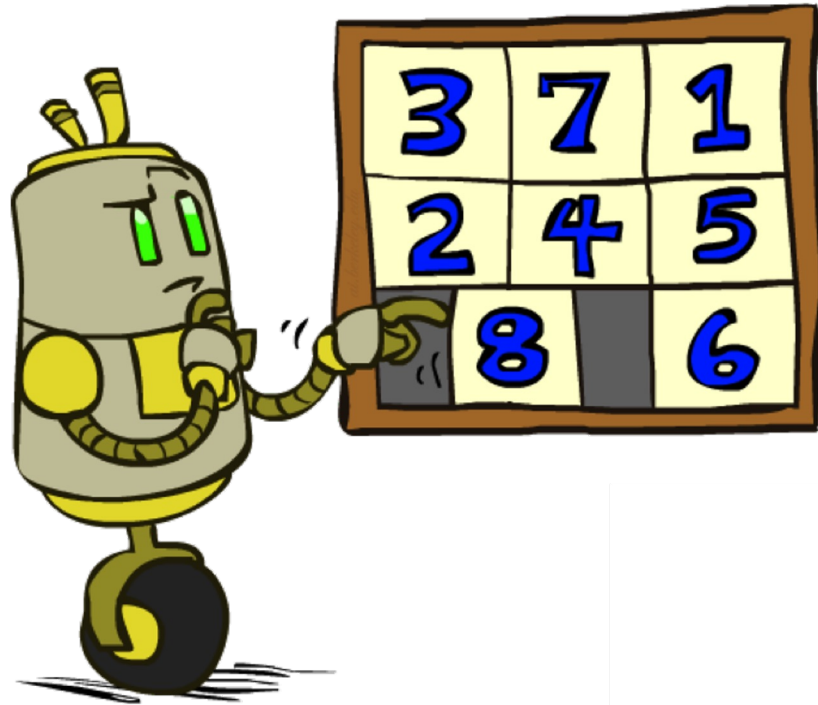


ایجاد هیوریستیک‌های قابل قبول برای پازل ۸ تایی



# ساده سازی مسئله پازل ۸ تایی

□ مسئله ساده شده (Relaxed Problem): نسخه ای از مسئله اصلی که در آن یک یا چند محدودیت حذف شده اند.

□ محدودیت ها: [برای حرکت دادن یک کاشی از خانه A به خانه B]

I. خانه B باید خالی باشد.

II. خانه B باید همسایه خانه A باشد.

□ مسئله ساده شده :

- نسخه ۱: هر دو محدودیت حذف شوند.
- نسخه ۲: محدودیت I حذف شود.
- نسخه ۳: محدودیت II حذف شود.

7	2	4
5		6
8	3	1

حالت شروع

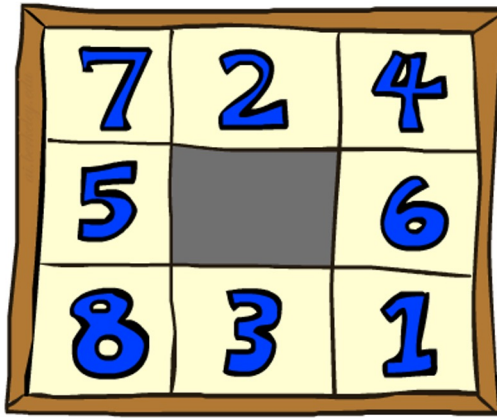
	1	2
3	4	5
6	7	8

حالت هدف

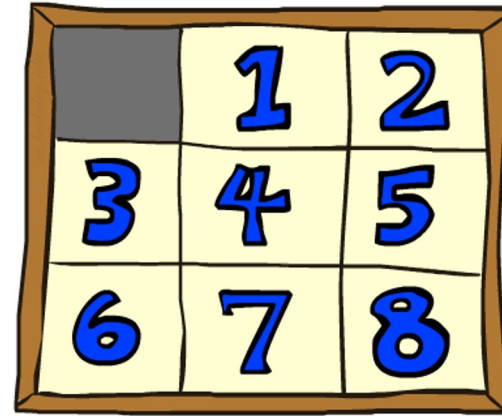
✓ هزینه دقیق یک مسئله ساده شده را می توان به عنوان تخمینی از هزینه واقعی مسئله اصلی در نظر گرفت.

# ساده سازی مسئله پازل ۸ تایی - ۱

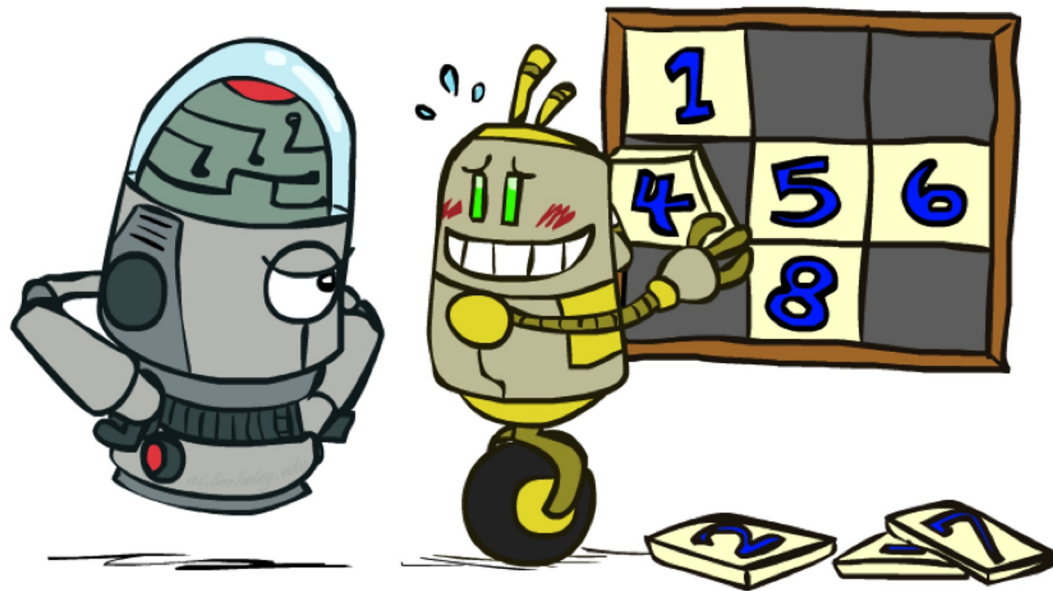
$h(\text{start}) = 8$



حالت شروع



حالت هدف



□ تابع هیوریستیک مسئله ساده شده، وقتی هر دو محدودیت زیر حذف شوند

I. خانه B باید خالی باشد.

II. خانه B باید همسایه خانه A باشد.

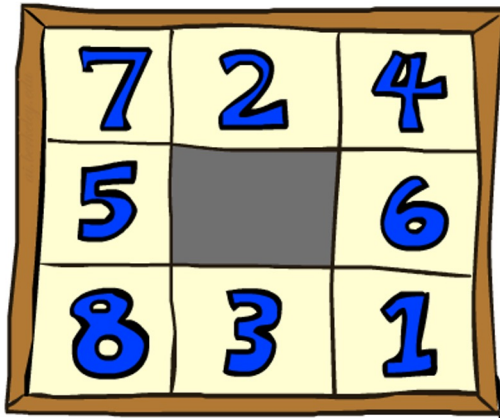
□ تابع هیوریستیک = تعداد کاشی هایی که در مکان نادرست قرار دارند.

Average nodes expanded  
when the optimal path has...

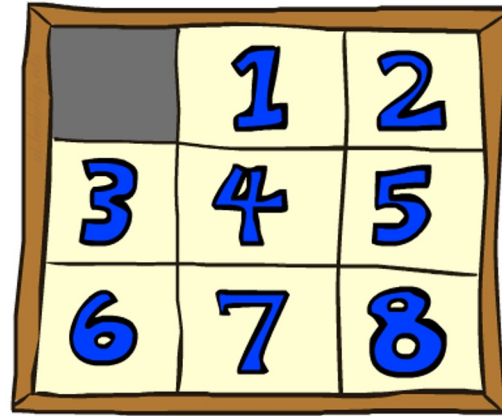
	...4 steps	...8 steps	...12 steps
UCS	112	6,300	$3.6 \times 10^6$
A*	13	39	227
TILES			3

## ساده سازی مسئله پازل ۸ تایی - ۲

$$h(\text{start}) = 3 + 1 + 2 + \dots = 18$$



حالت شروع



حالت هدف

□ تابع هیوریستیک مسئله ساده شده، وقتی فقط محدودیت زیر حذف شوند

I. خانه B باید خالی باشد.

□ تابع هیوریستیک = مجموع فواصل منهتنی کاشی ها تا مکان هدف.

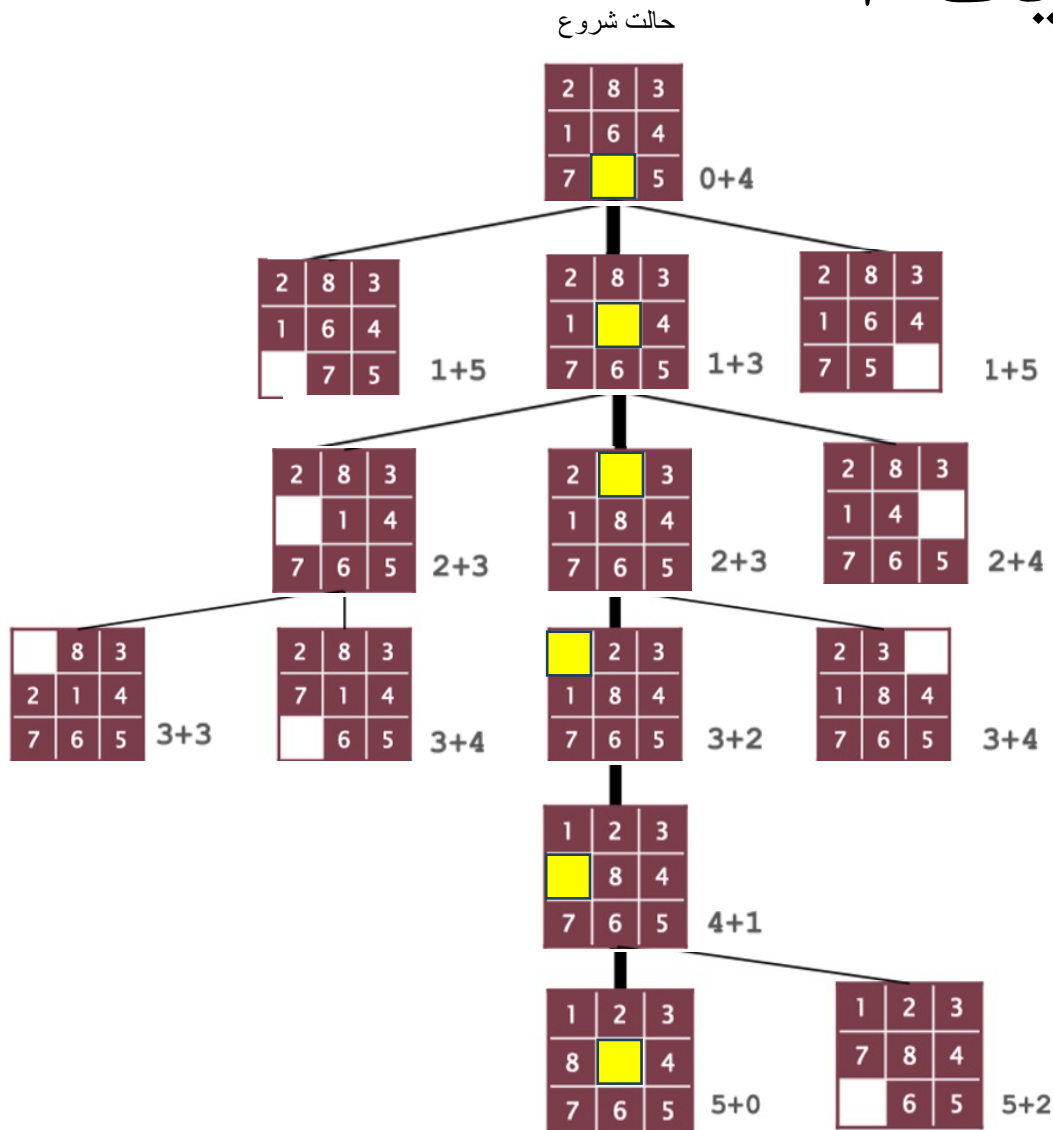
Average nodes expanded when the optimal path has...			
	...4 steps	...8 steps	...12 steps
TILES	13	39	227
A*	12	25	73
MANHATTAN			

# مثال: پازل ۸ تایی با تابع هیوریستیک ۱

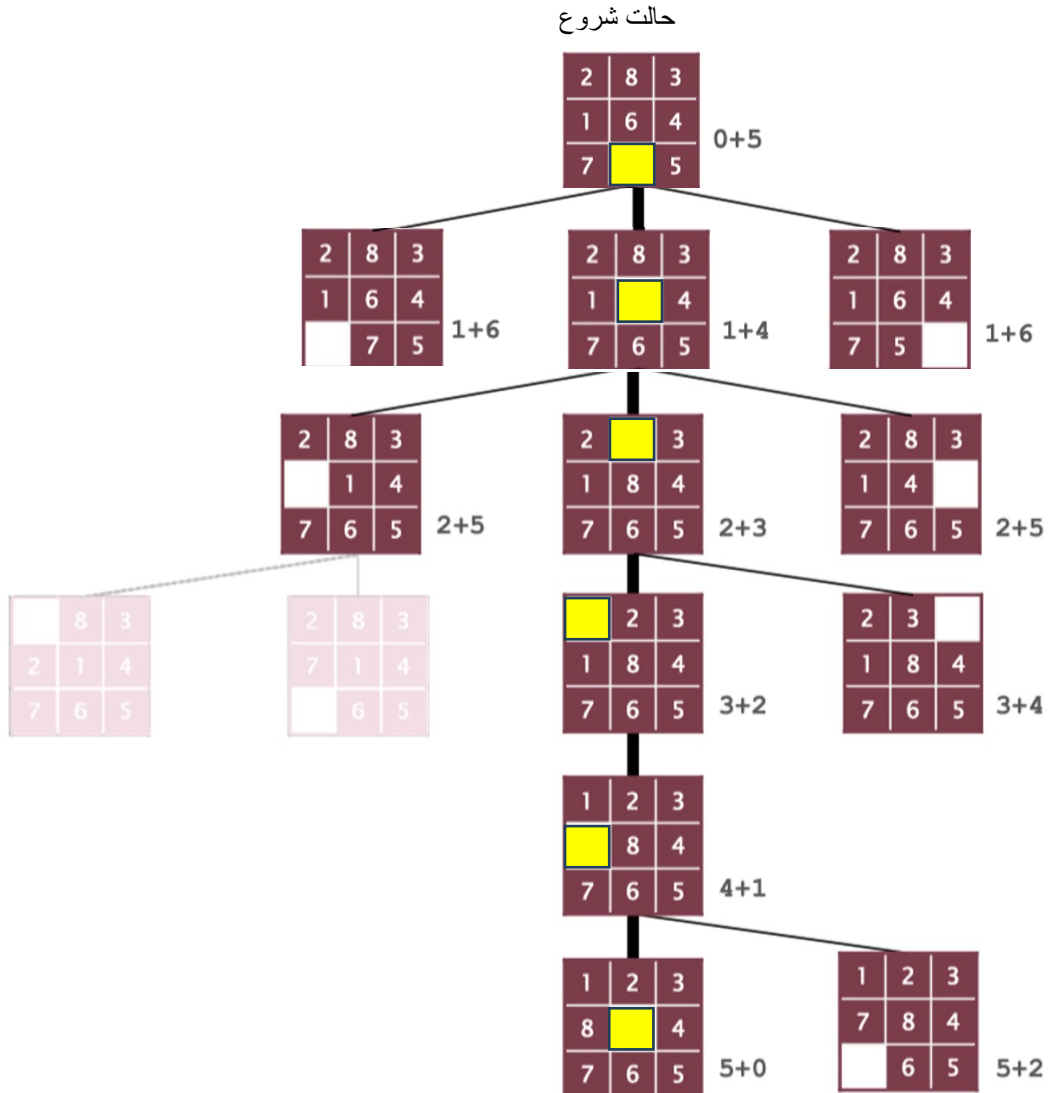
□ تابع هیوریستیک = تعداد کاشی‌هایی که در مکان نادرست قرار دارند.

□ حالت هدف

□ راه حل: خانه خالی باید به صورت: [بالا، بالا، چپ، پایین، راست] جابه‌جا شود



## مثال: پازل ۸ تایی با تابع هیوریستیک ۲



□ تابع هیوریستیک = مجموع فواصل  
منهتنی کاشی‌ها تا مکان هدف.

□ حالت هدف

□ راه حل: خانه خالی باید به صورت: [بالا،  
بالا، چپ، پایین، راست] جابه‌جا شود

# ترکیب توابع هیوریستیک

□ تسلط (Dominance): اگر به ازای هر  $n$  داشته باشیم:

$$h_2(n) \geq h_1(n) \quad (\text{الف})$$

(ب) و هر دو تابع هیوریستیک قابل قبول باشند،

آن گاه  $h_2$  بر  $h_1$  تسلط دارد.

## □ نکات کلیدی:

- تابع هیوریستیک با مقدار بزرگ‌تر، بهتر است، تا زمانی که هر دو قابل قبول باشند.
- تابع هیوریستیک صفر (UCS) بسیار ضعیف است (هر هیوریستیک دیگر بر آن تسلط دارد)
- استفاده از هزینه واقعی به عنوان یک تابع هیوریستیک ( $h = h^*$ ) خیلی خوب است و بهینه است اما معمولاً بسیار پرهزینه است

## □ ترکیب توابع هیوریستیک:

- اگر دو تابع هیوریستیک هیچ کدام بر دیگری تسلط نداشته باشند، می‌توان با ترکیب آن‌ها یک تابع هیوریستیک جدید تعریف کرد:  $h(n) = \max(h_1(n), h_2(n))$
- حداکثر مقدار دو تابع هیوریستیک قابل قبول، همچنان قابل قبول است و بر هر دو غلبه دارد!



# جستجوی $A^*$ : خلاصه

$$f(n) = g(n) + h(n)$$

- ❑ جستجوی  $A^*$  هم از هزینه پیموده شده تا الان استفاده می کند و هم از پیش بینی هزینه از الان تا هدف.
- ❑ جستجوی  $A^*$  برای جستجوی درختی بهینه است اگر از توابع هیوریستیک قابل قبول استفاده کند.
- ❑ طراحی تابع هیوریستیک یک نکته کلیدی است که اغلب برای طراحی آن از مسئله ساده شده (Relaxed Problem) استفاده می شود.



جستجوی هزینه یکنواخت



جستجوی فریبانه



جستجوی  $A^*$



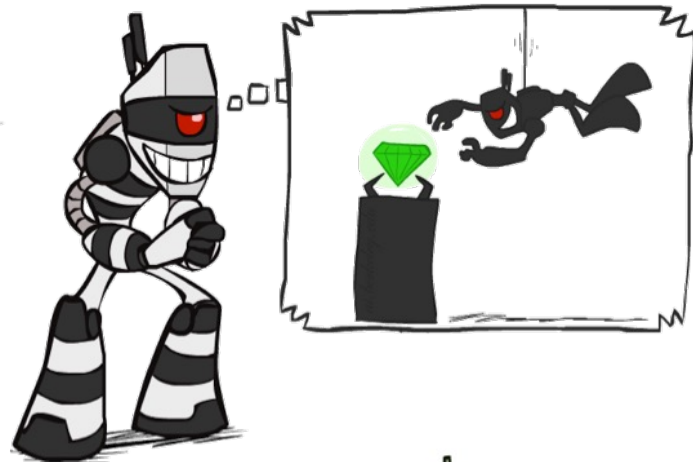
# مسائل ارضای محدودیت

Constraint Satisfaction Problems (CSP)



# جستجو برای چیست؟

❑ فرضیات درباره محیط. تک عاملی، قطعی، کاملاً قابل مشاهده، گسسته



❑ برنامه‌ریزی: یافتن دنباله‌ای از عملیات

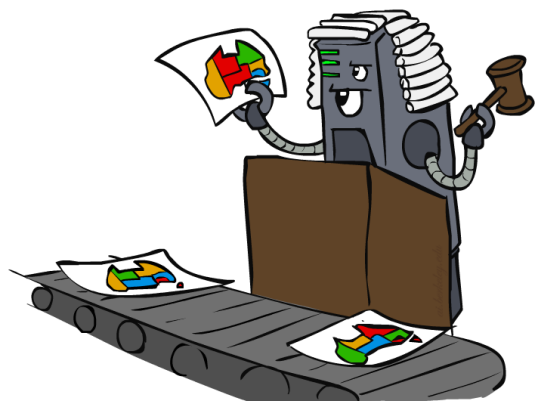
- مسیر یافته شده تا هدف مهمترین موضوع است.
- مسیرهای مختلف دارای هزینه‌ها و عمق‌های متفاوتی هستند.
- هیوریستیک‌ها مسیر جستجو را به سمت هدف هدایت می‌کنند.

❑ شناسایی: انتساب مقدار به متغیرها

- خود هدف مهم است و نه مسیر رسیدن به هدف.
- همه مسیرها دارای عمق یکسانی هستند.
- جستجوی ارضای محدودیت مختص مسائل شناسایی است.



# مسائل ارضای محدودیت

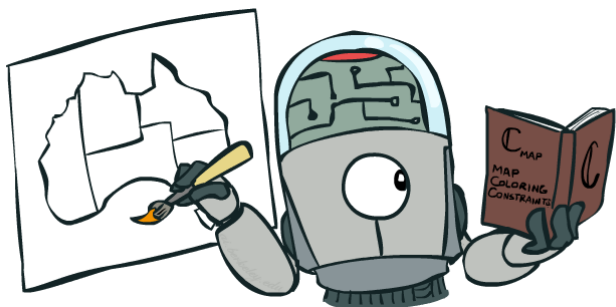


□ مسائل جستجوی استاندارد:

- حالت‌ها به صورت «جعبه سیاه» هستند: ساختمان داده‌های دلخواه
- تابع آزمون هدف، هر تابعی می‌تواند باشد.
- تابع جانشین نیز هر تابعی می‌تواند باشد.

□ مسائل ارضای محدودیت:

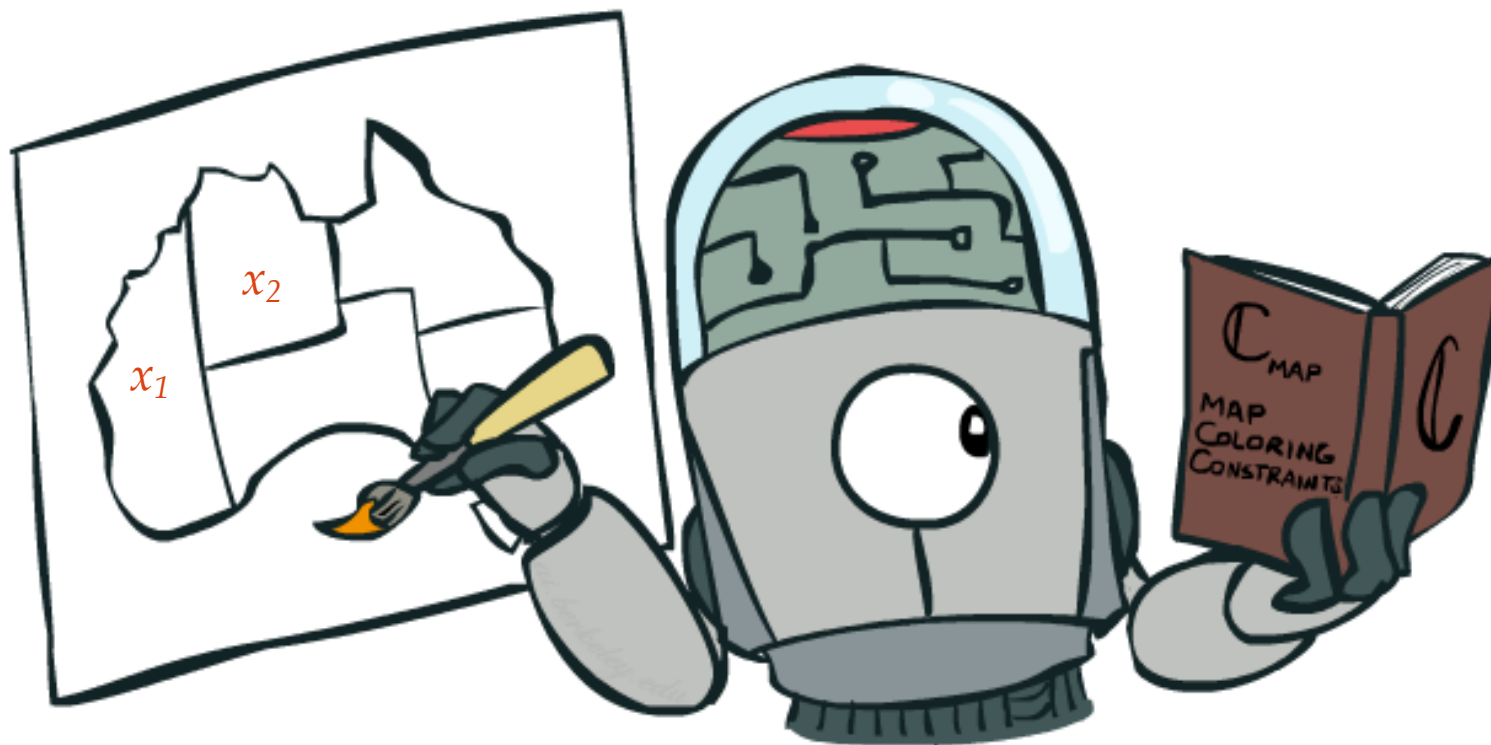
- یک زیرمجموعه خاص از مسائل جستجو
- حالت به وسیله مجموعه‌ای از **متغیرهای**  $X_i$  تعریف می‌شود که مقدار خود را از یک **دامنه**  $D_i$  می‌گیرند
- آزمون هدف یک **مجموعه از محدودیت‌ها** است که یک ترکیب مجاز از مقادیر را برای متغیرها مشخص می‌کنند.



✓ امکان استفاده از الگوریتم‌های عمومی مفید را فراهم می‌کند که قدرت بیشتری نسبت به الگوریتم‌های جستجوی استاندارد دارند.

# مسائل ارضای محدودیت

*N variables*  
*domain D*  
*constraints*



*states*  
*partial assignment*

*goal test*  
*complete; satisfies constraints*

*successor function*  
*assign an unassigned variable*

# تعریف CSP ها (مسائل ارضای محدودیت ها)

□ یک CSP از سه مؤلفه  $X$ ،  $D$  و  $C$  تشکیل شده است:

▪  $X$  مجموعه ای از متغیرها است:  $\{X_1, \dots, X_n\}$

▪  $D$  مجموعه ای از دامنه ها است:  $\{D_1, \dots, D_n\}$  به ازای هر متغیر یک دامنه

• یک دامنه  $D_i$  شامل مجموعه ای از مقادیر مجاز  $\{v_1, \dots, v_k\}$  برای متغیر  $X_i$  است.

▪  $C$  مجموعه ای از محدودیت ها است که ترکیب های مجاز از مقادیر را مشخص می کند

• هر محدودیت  $C_i$  شامل یک جفت  $\langle scope, rel \rangle$  است

•  $scope$  یا دامنه مجموعه ای از متغیرها است که در محدودیت شرکت می کنند

$\langle (X_1, X_2), X_1 > X_2 \rangle$  or  $\langle (X_1, X_2), \{(3, 2), (3, 1), (2, 1)\} \rangle$

•  $rel$  یا رابطه مشخص می کند که این متغیرها چه مقادیری می توانند بگیرند

# تعریف CSP ها (مسائل ارضای محدودیت ها)

□ CSP ها با انتساب مقادیر به متغیرها سروکار دارند،  $\{X_i = v_i, \dots, X_j = v_j\}$

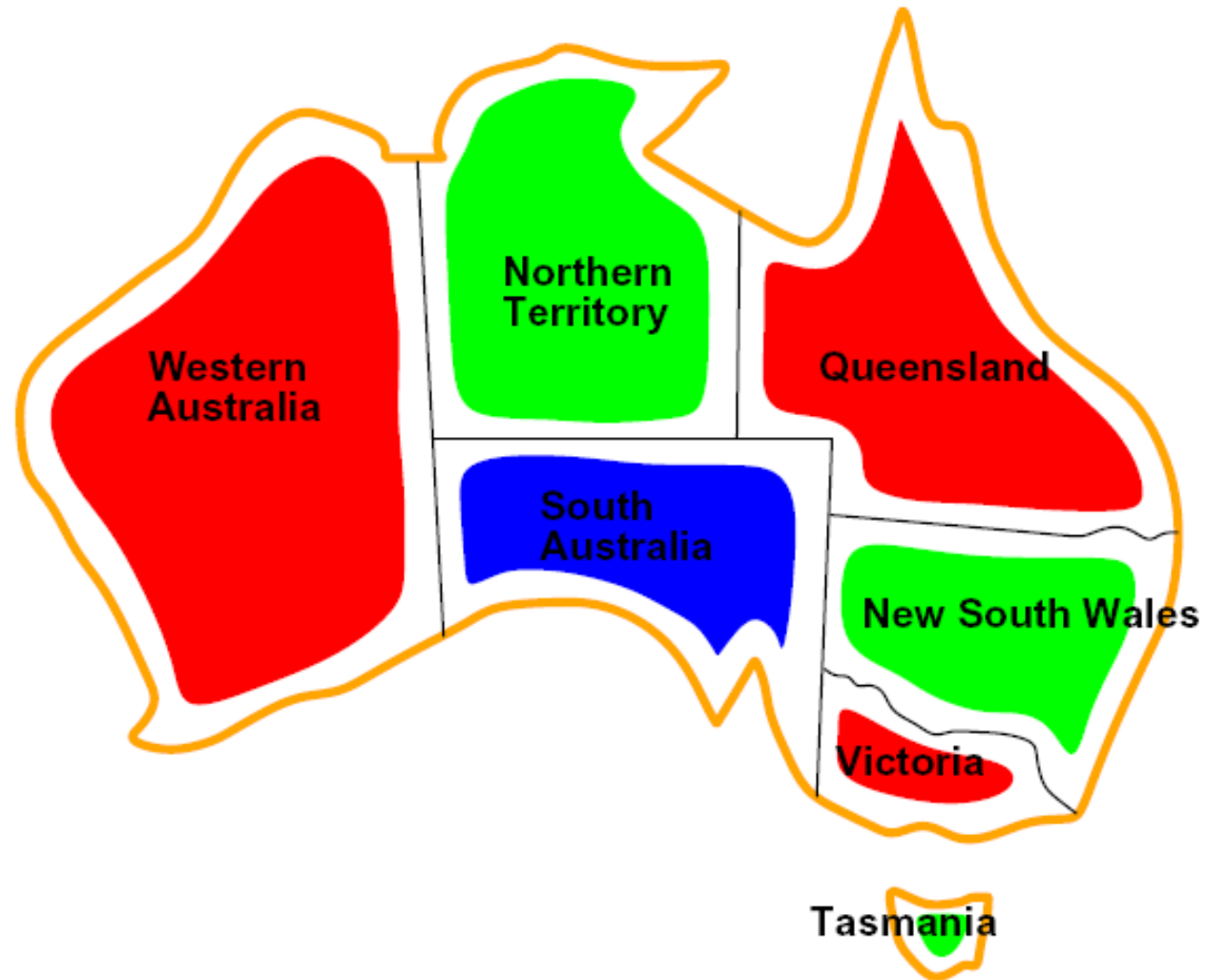
□ یک انتساب جزئی، انتسابی است که برخی متغیرها بدون مقدار باقی می ماند.

□ یک راه حل برای CSP، یک انتساب کامل و سازگار است.

□ یک انتساب کامل، انتسابی است که در آن هر متغیر مقداردهی شده باشد.

□ یک انتساب که هیچ یک از محدودیت ها را نقض نکند، سازگار یا یک انتساب قانونی نامیده می شود.

# مثال‌های CSP





# مثال: رنگ آمیزی نقشه

متغیرها:  $X = \{WA, NT, Q, NSW, V, SA, T\}$

دامنه‌ها:  $D = \{\text{Red, Green, Blue}\}$

محدودیت‌ها:

• ضمنی:  $WA \neq NT$  شهرهای مجاور باید رنگ‌های متفاوتی داشته باشند

• صریح:  $(WA, NT) \in \{(\text{red, green}), (\text{red, blue}), \dots\}$

راه حل‌ها: انتساب‌هایی هستند که تمام متغیرها مقدار بگیرند، در حالیکه همه محدودیت‌ها را برآورده کنند، (یک انتساب کامل و سازگار)  
مثال:  $\{WA=\text{red}, NT=\text{green}, Q=\text{red}, NSW=\text{green}, V=\text{red}, SA=\text{blue}, T=\text{green}\}$

