

ICMC-USP
Lista de Exercícios Capítulo 3 - Satisfação de Restrições
IA1 - SCC-5774

1o. Semestre de 2020 - Prof. João Luís Rosa

1. Quantas soluções existem para o problema de coloração de mapas da Fig. 1? Quantas soluções se quatro cores forem permitidas? E duas cores?

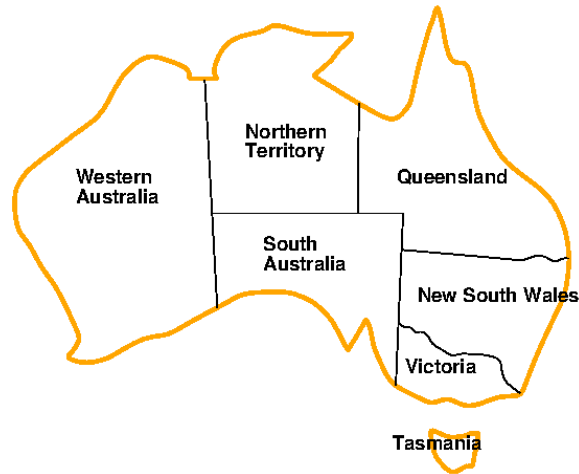


Figura 1: Os principais estados e territórios da Austrália [1].

2. Considere o problema de colocar k cavalos em um tabuleiro de xadrez $n \times n$ tal que os cavalos não se ataquem, onde k é dado e $k \leq n^2$.
- Escolha uma formulação CSP. Quais são as variáveis?
 - Quais são os valores possíveis para cada variável?
 - Quais conjuntos de variáveis são restritos, e como?
 - Agora considere o problema de colocar *o máximo número de cavalos possível* no tabuleiro sem nenhum ataque. Explique como resolver com busca local definindo funções AÇÕES e RESULTADO e uma função objetivo sensível.
3. Considere o problema de construir (não resolver) palavras-cruzadas: colocando palavras em uma grade retangular. A grade, que é dada como parte do problema, especifica quais quadrados são brancos e quais são sombreados. Assuma que uma lista de palavras (i.e. um dicionário) é provido e que a tarefa é preencher os quadrados em branco usando qualquer subconjunto da lista. Formule este problema precisamente de duas formas:
- Como um problema de busca geral. Escolha um algoritmo de busca apropriado e especifique uma função heurística. É melhor preencher os brancos com uma letra ou uma palavra por vez?
 - Como um problema de satisfação de restrições. Devem as variáveis ser palavras ou letras?

Na sua opinião, qual formulação será melhor? Por que?

4. Dê as formulações precisas para cada um dos seguintes problemas como problemas de satisfação de restrições:
 - a. Planejamento de piso retilíneo: encontrar locais não sobrepostos em um retângulo grande para um número de retângulos menores.
 - b. Agendamento de aulas: há um número fixo de professores e salas de aula, uma lista de classes a serem oferecidas e uma lista de possíveis horários para as aulas. Cada professor tem um conjunto de aulas que pode ensinar.
 - c. Turnê hamiltoniana: dada uma rede de cidades conectadas por estradas, escolha uma ordem para visitar todas as cidades em um país sem repetir nenhuma.
5. Resolva o problema criptaritmético da Fig. 2 manualmente, usando a estratégia de *back-tracking* com verificação progressiva (*forward checking*) e o MRV e heurística de menor valor de restrição.

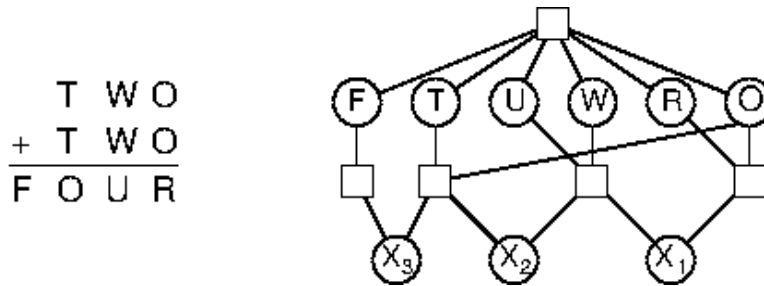


Figura 2: Hipergrafo de restrição.

6. Mostre como uma única restrição ternária, como “ $A + B = C$ ” pode ser transformada em três restrições binárias usando uma variável auxiliar. Você pode assumir domínios finitos. (*Dica:* Considere uma nova variável que assume valores que são pares de outros valores e considere restrições como “X é o primeiro elemento do par Y.”) Em seguida, mostre como as restrições com mais de três variáveis podem ser tratadas de forma semelhante. Finalmente, mostre como as restrições unárias podem ser eliminadas alterando os domínios das variáveis. Isto completa a demonstração de que qualquer CSP pode ser transformado em um CSP com apenas restrições binárias.

ICMC-USP
Lista de Exercícios - Capítulo 3
SCC-5774 (continuação)

7. Considere o seguinte enigma lógico: Em cinco casas, cada uma com uma cor diferente, vivem cinco pessoas de diferentes nacionalidades, cada uma prefere uma marca diferente de doce, uma bebida diferente e um animal de estimação diferente. Tendo em conta os seguintes fatos, as perguntas a responder são “Onde mora a zebra e em qual casa eles bebem água?”

O inglês mora na casa vermelha.
O espanhol é dono do cachorro.
O norueguês vive na primeira casa à esquerda.
A casa verde é imediatamente à direita da casa de marfim.
O homem que come barras Hershey vive na casa ao lado do homem com a raposa.
Kit Kats são comidos na casa amarela.
O norueguês mora ao lado da casa azul.
O comedor de Smarties possui caracóis.
O comedor de Snickers bebe suco de laranja.
O ucraniano bebe chá.
O japonês come Milky Ways.
Kit Kats são comidos em uma casa ao lado da casa onde o cavalo é mantido.
O café é bebido na casa verde.
O leite é bebido na casa do meio.

Discuta diferentes representações deste problema como um CSP. Por que alguém preferiria uma representação em relação à outra?

8. Explique por que é uma boa heurística escolher a variável que é *a mais* restrita, mas o valor que é *menos* restritivo em uma busca CSP.
9. Use o algoritmo AC-3 para mostrar que a consistência da aresta pode detectar a inconsistência da atribuição parcial $\{WA = verde, V = vermelho\}$, para o problema mostrado na Fig. 1.
10. Qual é a complexidade do pior caso ao executar o AC-3 em um CSP estruturado em árvore?
11. AC-3 coloca de volta na fila *toda* aresta (X_k, X_i) sempre que *qualquer* valor for excluído do domínio de X_i , mesmo se cada valor de X_k é consistente com vários valores restantes de X_i . Suponha que, para cada aresta (X_k, X_i) , acompanhamos o número de valores restantes de X_i que são consistentes com cada valor de X_k . Explique como atualizar esses números de forma eficiente e, portanto, mostrar que a consistência da aresta pode ser aplicada no tempo total $\mathcal{O}(n^2d^2)$.

12. O TREE-CSP-SOLVER (Fig. 3) torna as arestas consistentes começando a partir das folhas e trabalhando para trás em direção à raiz. Por que ele faz isso? O que aconteceria se fosse na direção oposta?

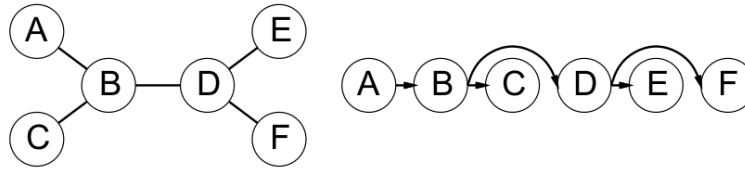


Figura 3: TREE-CSP-SOLVER.

13. Introduzimos o Sudoku como um CSP a ser resolvido pela busca sobre atribuições parciais porque é assim que as pessoas geralmente se comprometem a resolver os problemas do Sudoku. Também é possível, claro, atacar esses problemas com busca local sobre atribuições completas. Quão bem um solucionador local usando a heurística de conflitos mínimos faz em problemas de Sudoku?
14. Defina com suas próprias palavras os termos restrição, busca de retrocesso (*backtracking*), consistência da aresta, conflitos mínimos e *cutset* de ciclo.
15. Suponha que um grafo seja conhecido por ter um *cutset* de ciclo de não mais que k nós. Descreva um algoritmo simples para encontrar um *cutset* de ciclo mínimo cujo tempo de execução não é muito mais do que $\mathcal{O}(n^k)$ para um CSP com n variáveis. Pesquise na literatura os métodos para encontrar *cutsets* de ciclo aproximadamente mínimos no tempo que é polinomial no tamanho do *cutset*. A existência de tais algoritmos torna o método de *cutset* de ciclo prático?

Referências

- [1] Russell, S., Norvig, P. *Artificial Intelligence - A Modern Approach*. 3rd. edition. Prentice Hall, 2010.