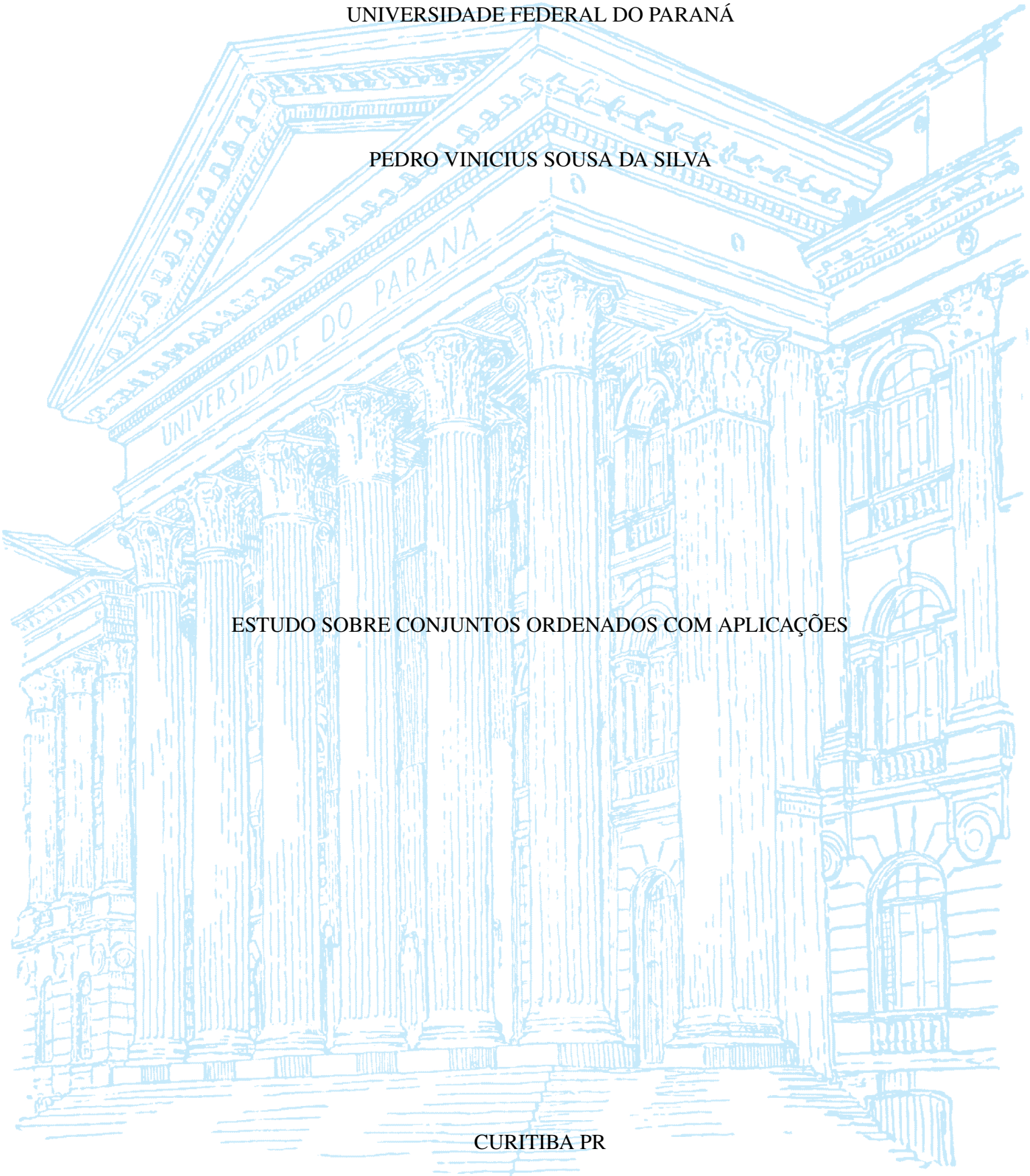
UNIVERSIDADE FEDERAL DO PARANÁ

PEDRO VINICIUS SOUSA DA SILVA



ESTUDO SOBRE CONJUNTOS ORDENADOS COM APLICAÇÕES

CURITIBA PR

2023

PEDRO VINICIUS SOUSA DA SILVA

ESTUDO SOBRE CONJUNTOS ORDENADOS COM APLICAÇÕES

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: André Luiz Pires Guedes.

CURITIBA PR

2023

Aos meus pais que sempre me ajuda-ram.

AGRADECIMENTOS

Ao Professor André Pires Guedes, por ter aceitado me orientar mesmo eu ainda bastante perdido. Aos meus pais pelas suas imensas dedicações em me proporcionar um bom estudo. As minhas irmãs que sempre me ajudaram cada uma do seu jeito. Aos amigos que me acompanharam nesta jornada. Ao Professor Renato Carmo pela sugestão do tema. Ao Professor Leandro Zatesko, ao Professor Murilo Vicente e ao Doutor Cleiton Almeida, pelas diversos comentários e sugestões para o trabalho.

RESUMO

O conceito de ordem está presente na maioria dos problemas computacionais, seu estudo permite um maior controle das propriedades do problema, e pode ajudar no desenvolvimento de algoritmos eficientes. Neste trabalho será dada uma introdução ao conceito de ordem utilizando o problema de encontrar a redução transitiva de um grafo e problemas de programação competitiva como motivação.

Palavras-chave: Conjuntos ordenados. Redução transitiva. Programação competitiva

ABSTRACT

The concept of order is present in most computational problems, its study allows a better control of the properties of the problem and can help in the development of efficient algorithms. This paper will give an introduction to the concept of order using the problem of finding the transitive reduction of a graph and competitive programming problems as motivation.

Keywords: Ordered sets. Transitive reduction. Competitive programming

LISTA DE FIGURAS

LISTA DE ACRÔNIMOS

DINF	Departamento de Informática
UFPR	Universidade Federal do Paraná
SBC	Sociedade Brasileira de Computação
POSET	*Partially Ordered Set*
LIS	*Longest Increasing Subsequence*
DAG	*Directed Acyclic Graph*

SUMÁRIO

1 INTRODUÇÃO

O presente trabalho teve como objetivo explorar de forma introdutória o conceito de ordem através do problema de encontrar a redução transitiva de um grafo e problemas de programação competitiva.

O problema da redução transitiva consiste em encontrar uma representação mais econômica de um grafo que mantenha a informação da existência de caminho entre dois vértices, perguntas como "Essa representação sempre existe?" e "Qual a complexidade de se computar tal representação?" serão respondidas no Capítulo 2.

A programação competitiva é um tipo de competição onde os participantes devem resolver um conjunto de problemas computacionais em um determinado limite de tempo. Cada problema de uma competição geralmente acompanha uma estória com a descrição do problema e especificações bem definidas para sua entrada e saída, com base nesses dados os participantes devem escrever uma programa, em alguma linguagem de programação disponível no ambiente da competição, que dê uma resposta correta para cada uma das entradas pré-estabelecidas. Os problemas de programação competitiva que serão vistos servirão como problemas de brinquedo para aplicar as definições e teoremas apresentados ao longo do texto.

No Capítulo 1, o conceito de ordem será formalizado e serão apresentados alguns resultados clássicos, como por exemplo, que a complexidade de encontrar o conjunto independente máximo do grafo representado por uma ordem, é polinomial. No Capítulo 2, será visto como encontrar a redução transitiva de um grafo direcionado, que no contexto de ordem, é conhecido como grafo subjacente do Diagrama de Hasse. Por fim, no Capítulo 3, serão mostradas aplicações dos conceitos vistos em problemas de programação competitiva.

2 CONJUNTOS ORDENADOS

2.1 INTRODUÇÃO

Desde crianças nos deparamos com o conceito de ordem, como por exemplo: através dos números naturais, das letras do alfabeto, das prioridades em executar tarefas do nosso cotidiano.

Uma ordem intuitivamente relaciona dois objetos utilizando suas propriedades, ao mesmo tempo que é específica para estes objetos, uma ordem não é intrínseca a eles, ou seja, diferentes ordens podem ser aplicadas em um mesmo conjunto de objetos. No contexto computacional, este fato dá origem ao problema de escolher qual a melhor ordem para um determinado problema.

Neste capítulo, será dada uma introdução à formalização matemática do conceito de ordem. Primeiro serão dadas as definições básicas. Em seguida, será mostrada uma maneira de representar visualmente uma ordem, e por fim serão descritos dois teoremas que serão úteis para a resolução dos problemas do Capítulo 4.

O texto usa como referência Davey e Priestley (2002), Might (2012), Trotter (2015) e Felsner (2020).

2.2 DEFINIÇÕES E NOTAÇÃO

Seja P um conjunto. Uma *ordem* em P é uma relação binária $\leq$ com as propriedades reflexiva, transitiva e antissimétrica.

Um *conjunto ordenado* $\mathbf{P}$ é um par $(P, \leq)$ onde P é um conjunto finito e $\leq$ é uma ordem sobre P.

Naturalmente a notação $x \geq y$ significa $y \leq x$.

Se $<$ for uma relação binária com as propriedades irreflexiva e transitiva, chamaremos $\mathbf{P} = (P, <)$ de *conjunto ordenado* com uma ordem estrita.

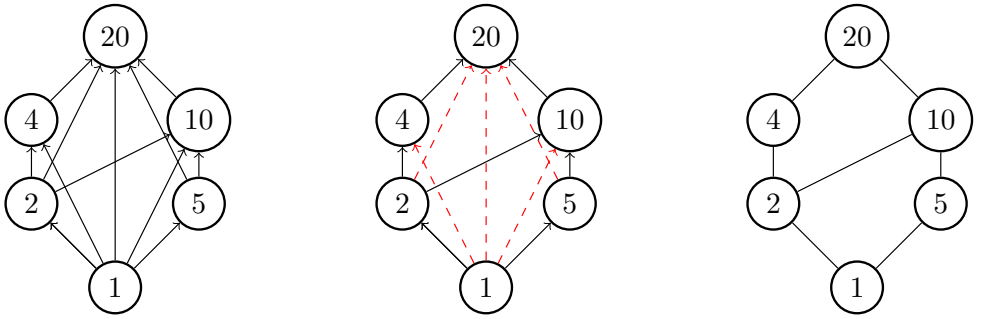
Dois elementos $x, y \in \mathbf{P}$, são comparáveis, denotado por $x \parallel y$, se $x \leq y$ ou $x \geq y$, e são incomparáveis caso contrário, denotado por $x \nparallel y$.

Um conjunto ordenado $\mathbf{P}$ é um *conjunto parcialmente ordenado* (ou *poset*) se possui elementos incomparáveis, por outro lado, $\mathbf{P}$ é uma cadeia (ou ordem linear ou ordem total) se todos os elementos de $\mathbf{P}$ são comparáveis, e de maneira dual $\mathbf{P}$ é uma anticadeia no caso de todos os elementos de $\mathbf{P}$ serem incomparáveis.

Seja $\mathbf{P} = (P, \leq)$ e $\mathbf{S} = (S, \leq')$ *posets*, onde $S \subseteq P$ e $\leq'$ é a relação $\leq$ restringida aos elementos de S, então $\mathbf{S}$ é definida como uma *subordem* de $\mathbf{P}$. Combinada com a definição anterior, se $\mathbf{S}$ também é uma (anti)cadeia, então $\mathbf{S}$ é uma (anti)cadeia de $\mathbf{P}$.

Alguns elementos de um *poset* $\mathbf{P} = (P, \leq)$ são importantes. Seja $x \in P$, x é um:

- elemento minimal de $\mathbf{P}$, se não existe $y \in P$, tal que $y \leq x$ e $y \neq x$.
- elemento maximal de $\mathbf{P}$ se não existe $y \in P$, tal que $y \geq x$ e $y \neq x$.
- elemento mínimo de $\mathbf{P}$, se $x \leq y, \forall y \in P$.
- elemento máximo de $\mathbf{P}$, se $x \geq y, \forall y \in P$.



(a) Representação prelimilar para $\boldsymbol{D_{20}}$ (b) Arcos redundantes destacadas (c) Diagrama de hasse para $\boldsymbol{D_{20}}$

Figura 2.1: Refinamento das representações visuais de $\boldsymbol{D_{20}}$

2.3 EXEMPLOS INICIAIS

1. Os conjuntos $\mathbb{R}, \mathbb{N}, \mathbb{Z}, \mathbb{Q}$ formam uma cadeia usando sua ordem usual.

2. Dado um conjunto S, onde 2^S é o conjunto das partes de S, temos que $\mathbf{P}$ é um *poset* com $P = 2^S$, tomando $\leq$ como a ordem por inclusão, formalmente dada por: $X \leq Y$ se e somente se $X \subseteq Y$.

3. Seja $n \in \mathbb{N}$, e D_n o conjunto dos divisores de n, $\boldsymbol{D_n}$ é um *poset* definido sobre D_n ordenado pela divisibilidade, ou seja, $x \leq y$ se e somente se x divide y.

2.4 DIAGRAMA DE HASSE

Um jeito natural de desenhar um conjunto ordenado $\mathbf{P} = (P, \leq)$ é utilizando uma representação baseada em grafos. Para isso, defina P como o conjunto de vértices e crie um arco (u, v) para cada $u \leq v$, onde $u, v \in P$. Esta representação é chamada de grafo subjacente de $\mathbf{P}$. Considerando o exemplo 3 da seção anterior, para $n = 20$ teríamos o grafo da Figura 2.1(a). Como podemos ver este grafo possui muitos arcos, o que dificulta sua visualização. Para resolver este problema, primeiro definimos um arco (u, v) como *redundante*, se existe um outro caminho de u para v que não utiliza o arco (u, v). Como veremos no Capítulo 3, o algoritmo de procurar por arcos redundantes e removê-los do grafo até que não haja mais arcos redundantes, nos leva a uma representação única para o tipo de grafo representado por $\mathbf{P}$. Por fim, para capturar a ideia que $u \leq v$ na representação visual, sem precisar indicar a direção, desenharemos u abaixo de v, extrapolando a ideia de grafo, esta representação é conhecida como Diagrama de Hasse. A Figura 2.1(b) destaca os arcos redundantes da representação do conjunto ordenado da Figura 2.1(a) e por fim, a representação final é mostrada em 2.1(c).

2.5 EXEMPLO (BALÃO)

Antes de avançarmos para os teoremas, pegaremos um exemplo de um problema de uma competição para aplicar os conceitos já visto.

Enunciado: Uma das principais dificuldades de organizar uma Maratona de Programação é recolher balões que escapam e ficam presos no teto do salão.

Este ano a organização da Maratona está mais previdente: ela tem o desenho do teto do salão, e quer sua ajuda para determinar o que pode acontecer com um balão, dependendo da posição no solo onde ele é solto (isto é, se é bloqueado pelo teto ou se escapa para o exterior do salão).

O teto do salão é formado por vários planos que, vistos de lado, podem ser descritos por segmentos de reta, como mostrado na Figura 2.2:

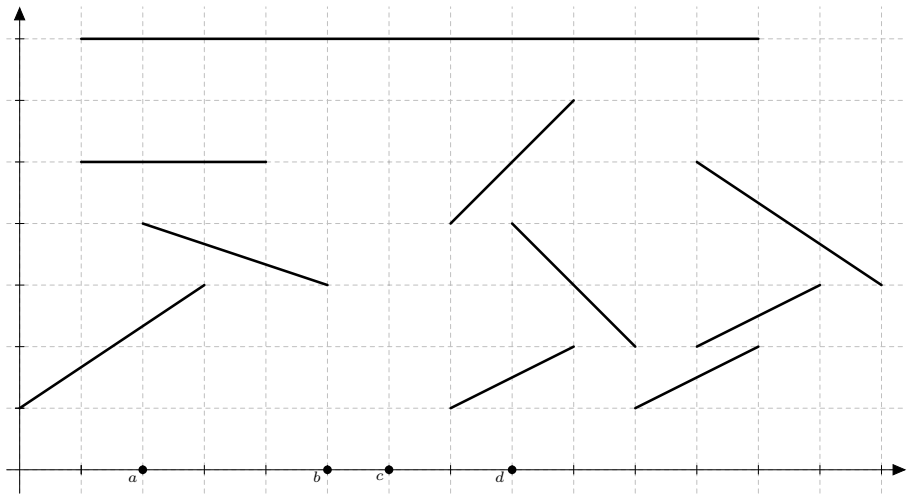


Figura 2.2: Telhado do salão visto de lado(Adaptado de SBC (2013))

O balão pode ser considerado como um ponto. Quando um balão toca um segmento do teto que é horizontal, ele fica preso. Quando um balão toca um segmento que é inclinado, o balão desliza até o ponto mais alto do segmento e escapa, podendo escapar completamente do salão ou podendo tocar em mais segmentos. Não há pontos em comum entre os segmentos que formam o teto.

Por exemplo, se o balão for solto nas posições marcadas como a ou b, será bloqueado na posição de coordenadas $(2, 5)$; se o balão for solto na posição marcada como c, será bloqueado na posição de coordenadas $(6, 7)$; e se o balão for solto na posição marcada como d será bloqueado na posição de coordenadas $(9, 7)$ (É garantido que para os balões dados na entrada sempre haverá um telhado que ele será bloqueado)

Escreva um programa que, dada a descrição do teto do salão como segmentos de reta, responde a uma série de consultas sobre a posição final de balões soltos do piso do salão.

(Adaptado de SBC (2013))

Para os nossos fins, focaremos apenas na relação entre os segmentos dispostos no plano.

Seja P o conjunto de segmentos que representam os telhados como o problema descreve e seja $\leq$ a relação definida entre dois segmentos u, v, onde $u \leq v$ se e somente se o balão que chega em u e eventualmente toca em v. É possível verificar que $\leq$ forma uma relação de ordem, e portanto podemos modelar o problema através de um *poset* $\mathbf{P} = (P, \leq)$. A Figura 2.3 mostra o Diagrama de Hasse para a entrada ilustrada no exemplo da Figura 2.2, nela os elementos maximais são destacados com a cor preta e os elementos minimais com a cor cinza [1].

2.6 ALTURA E LARGURA DE UMA ORDEM PARCIAL

A altura de um conjunto ordenado $\mathbf{P}$ é definida como o tamanho de sua maior cadeia, denotada por $h(\mathbf{P})$, e sua largura definida como o tamanho de sua maior anticadeia, denotada por $w(\mathbf{P})$. A Figura 2.4 mostra um exemplo deste conceito.

[1] Apenas para dar um vislumbre da solução, a ideia seria notar que particionar o *poset* da Figura 2.3 em cadeias é possível utilizando a técnica *Sweep line*, o que permite resolver o problema de maneira *offline*

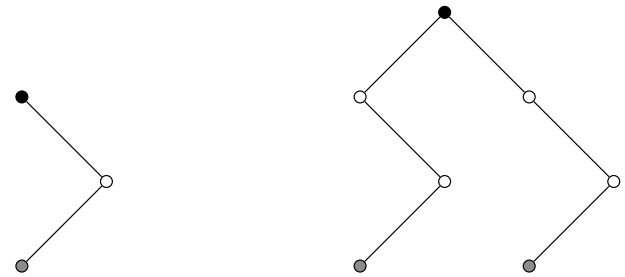
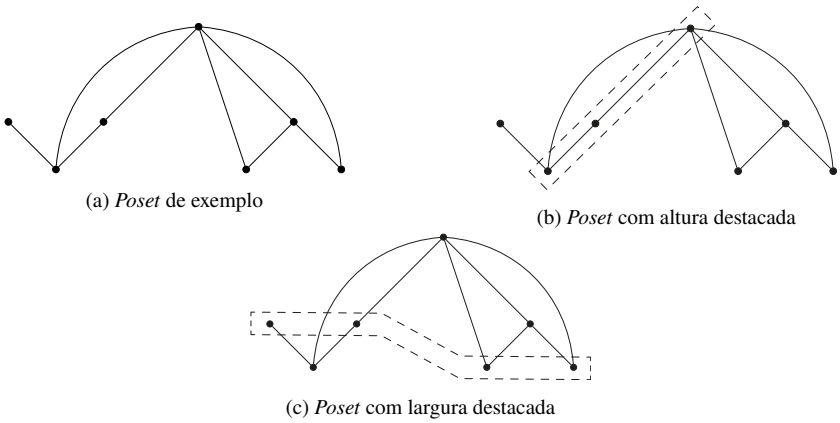


Figura 2.3: Diagrama de Hasse para uma entrada da Figura 2.2



(a) *Poset* de exemplo

(b) *Poset* com altura destacada

(c) *Poset* com largura destacada

Figura 2.4: Diagrama de hasse de um *poset* $\mathbf{P}$ com $h(\mathbf{P}) = 3$ e $w(\mathbf{P}) = 4$

2.7 PARTIÇÕES

Uma k-partição de um *poset* $\mathbf{P} = (P, \leq)$ em (anti)cadeias, é definida como um conjunto de k (anti)cadeias de $\mathbf{P}$, $\boldsymbol{S_i}$ $(1 \leq i \leq k)$, onde $S_1, S_2, \cdots, S_k$ são conjuntos dois a dois disjuntos e $\bigcup_{1 \leq i \leq k} S_i = P$.

TEOREMA 1 (Teorema de Mirsky (1971))**.** *Em um poset, o número mínimo de anticadeias necessárias para particioná-lo, é igual ao tamanho de sua maior cadeia.*

Demonstração. Seja $a(\mathbf{P})$ ≔ número de mínimo de anticadeias necessárias para particionar os elementos de $\mathbf{P}$.

Dessa maneira a demonstração será dividida em mostrar que:

- $a(\mathbf{P}) \geq h(\mathbf{P})$
- $a(\mathbf{P}) \leq h(\mathbf{P})$

($a(\mathbf{P}) \geq h(\mathbf{P})$) Utilizando o Princípio da casa dos pombos temos que $a(\mathbf{P}) \geq h(\mathbf{P})$, pois caso $a(\mathbf{P}) < h(\mathbf{P})$, pelo menos dois elementos da maior cadeia estariam em uma mesma partição, o que contradiz o fato de estarmos particionando em anticadeias.

($a(\mathbf{P}) \leq h(\mathbf{P})$) A ideia intuitiva é notar que os elementos minimais de $\mathbf{P}$ formam uma anticadeia, e dessa maneira podemos sucessivamente remover os elementos minimais e mantê-los em partições diferentes.

Como neste processo não podemos ter mais que $h(\mathbf{P})$ elementos minimais removidos, temos que $a(\mathbf{P}) \leq h(\mathbf{P})$. □

TEOREMA 2 (Teorema de Dilworth (1950))**.** *O número mínimo de cadeias necessárias para particionar os elementos de* $\boldsymbol{P}$ *é igual ao tamanho da maior anticadeia de* $\boldsymbol{P}$*.*

Demonstração. Seja $c(\mathbf{P})$:= número mínimo de cadeias necessárias para particionar os elementos de $\mathbf{P}$.

Similar ao teorema anterior podemos dividir a demonstração em mostrar que:

- $c(\mathbf{P}) \geq w(\mathbf{P})$
- $c(\mathbf{P}) \leq w(\mathbf{P})$

($c(\mathbf{P}) \geq w(\mathbf{P})$) Nesse caso, também podemos fazer uso do Princípio da casa dos pombos, pois supondo que $c(\mathbf{P}) < w(\mathbf{P})$ teríamos pelo menos dois elementos da maior anticadeia em uma mesma partição, o que contradiz o fato que estarmos particionando em cadeias.

($c(\mathbf{P}) \leq w(\mathbf{P})$) Para esse caso, será apresentada uma demostração com base no Teorema de König[2]. Esta demonstração segue os argumentos descritos em Goemans (2004).

TEOREMA 3 (Teorema de König)**.** *Para qualquer grafo bipartido, o tamanho do emparelhamento máximo é igual ao tamanho da cobertura mínima por vértices.* □

Para mostrarmos $c(\mathbf{P}) \leq w(\mathbf{P})$ para $\mathbf{P}$, iremos reduzir uma partição em cadeias em um emparelhamento e uma anticadeia em uma cobertura de um grafo bipartido.

Para isso, seja H um grafo bipartido com $2n$ vértices ($n = |P|$), onde para cada $x \in P$, criamos dois vértices u_x e v_x, e o conjunto de arestas de H é tal que $\{u_x, v_y\} \in E(H) \Leftrightarrow x \leq y$ e $x \neq y$, a Figura 2.6 ilustra essa ideia.

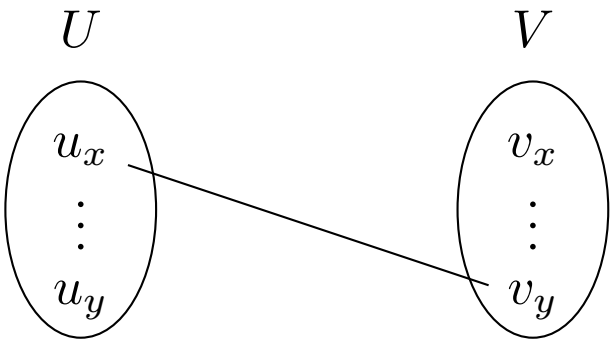


Figura 2.5: Grafo bipartido usado na redução

Agora, considere um emparelhamento M do grafo H, temos que esse emparelhamento corresponde a uma partição de S com $n - |M|$ cadeias. Isso acontece pois, se consideramos apenas as arestas do emparelhamento, cada vértice u_x que está presente nesse emparelhamento, continua uma cadeia para um único outro elemento maior que ele. Por fim, para contarmos quantas cadeias foram criadas basta contar quantos são os elementos minimais, e esse valor é igual a quantidade de vértices em V que não foram emparelhados, ou seja, $n - |M|$. As Figuras 2.6 e 2.7 mostram um exemplo dessa redução.

Voltando ao que queríamos provar, considere F uma cobertura mínima por vértices de H. Seja $A = \{x \in P \quad | \quad u_x \notin F \text{ e } v_x \notin F\}$. O fato de F ser uma cobertura mínima implica que A é uma anticadeia. Ainda mais, temos que $|A| \geq n - |F| \overset{Konig}{=} n - |M| = c(\mathbf{P})$. □

TEOREMA 4 (Teorema de Mirsky Generalizado (Goemans (2004)))**.** *Em um poset* $\boldsymbol{P}= (P, \leq)$ *com uma função de pesos* $w : P \to \mathbb{Z}_+$ *, o número mínimo de anticadeias necessárias para particioná-lo, onde cada* $x \in P$ *aparece em* $w(x)$ *partes, é igual a cadeia de maior soma de pesos.*

[2]Seria possível ir por um racícionio mais fácil, usando um argumento indutivo, porém como no Capítulo 4 serão estudadas aplicações desse teorema é útil ver uma demonstração mais construtiva.

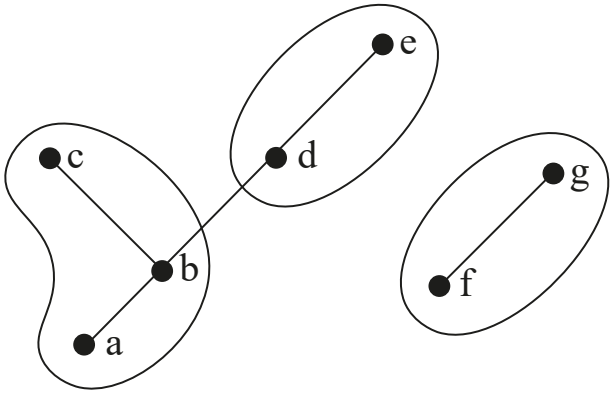


Figura 2.6: Exemplo de uma particao em cadeias de um *poset*.

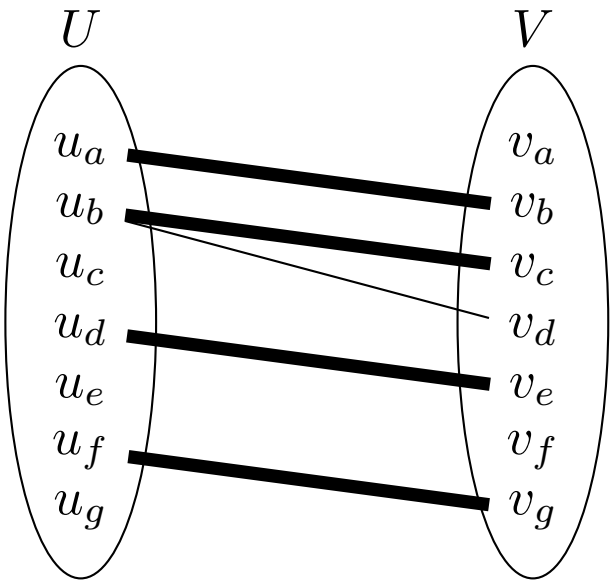


Figura 2.7: Grafo bipartido com o emparelhamento destacado, este emparelhamento correspondente à particao em cadeias do *poset* da Figura 2.6. Note que os elementos minimais desta partição, correspondem aos elementos que estão em V e não estão no emparelhamento

Demonstração. Substitua cada vértice x por uma cadeia com $w(x)$ cópias de x, $\forall x \in P$. Se $x \leq y$ no *poset* original, então $x^{(i)} \leq y^{(j)}$ onde $x^{(i)}$ é qualquer cópia de x, analogamente para $y^{(j)}$. Veja a Figura 2.8. Seja **P'** o *poset* resultante. Então aplique o Teorema de Mirsky em **P'**. Uma cadeia de tamanho máximo em **P'** corresponde a uma cadeia de maior soma de pesos em **P**. De maneira similar, uma partição mínima de **P'** em anticadeias corresponde em uma partição mínima de **P** em anticadeias, onde cada $x \in P$ aparece em $w(x)$ partes.

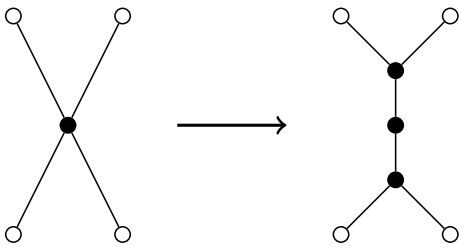


Figura 2.8: Expansão de um elemento em uma cadeia de cópias de si mesmo.

□

TEOREMA 5 (Teorema de Dilworth Generalizado (Goemans (2004))). *Em um poset* $\boldsymbol{P} = (P, \leq)$ *com uma função de pesos* $w : P \to \mathbb{Z}_+$ *, o número mínimo de cadeias necessárias para particioná-lo, onde cada* $x \in P$ *aparece em* $w(x)$ *partes, é igual a anticadeia de maior soma de pesos.*

Demonstração. O esquema da demonstração anterior se repete. Substitua cada vértice x por uma anticadeia com $w(x)$ cópias de x, $\forall x \in P$. Se $x \leq y$ no *poset* original, então $x^{(i)} \leq y^{(j)}$ onde $x^{(i)}$ é qualquer cópia de x, analogamente para $y^{(j)}$. Veja a Figura 2.9. Seja **P'** o *poset* resultante. Então aplique o Teorema de Dilworth em **P'**. Uma anticadeia de tamanho máximo em **P'** corresponde a uma anticadeia de maior soma de pesos em **P**. De maneira similar, uma partição mínima de **P'** em cadeias corresponde em uma partição mínima de **P** em cadeias, onde cada $x \in P$ aparece em $w(x)$ partes.

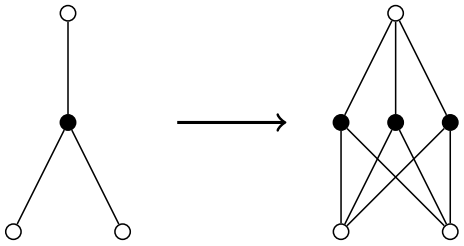


Figura 2.9: Expansão de um elemento em uma anticadeia de cópias de si mesmo.

□

3 REDUÇÃO TRANSITIVA

3.1 INTRODUÇÃO

Neste capítulo será visto uma maneira de encontrar uma representação mais econômica (com menos arcos) de um grafo direcionado G, que mantenha a informação da existência de caminho entre dois vértices. O texto segue os resultados descritos em Aho et al. (1972). Inicialmente serão mostrados os teoremas que demonstram que tal representação existe e é única, quando o grafo direcionado é acíclico, a partir desse resultado o próximo passo será adaptar os teoremas para lidar com o caso geral, por fim será visto o algoritmo para computar esta representação em tempo polinomial, através de uma redução ao problema do fecho transitivo.

3.2 DEFINIÇÕES E NOTAÇÃO

Nesta seção serão apresentadas algumas definições que serão importantes no restante do texto, com exceção da definição de grafo direcionado, as outras definições seguem as definições clássicas descritas em Bondy (2008).

DEFINIÇÃO 3.2.1 (Grafo direcionado)**.** Um *grafo direcionado* G sobre um conjunto de vértices $V = \{v_1, v_2, \ldots, v_n\}$ é um subconjunto de $V \times V$.

DEFINIÇÃO 3.2.2 (Grafo transitivo)**.** Um grafo G é dito transitivo se, para todo par de vértices u e v, não necessariamente distintos, $(u, v) \in G$ sempre que há um caminho em G de u para v.

DEFINIÇÃO 3.2.3 (Fecho transitivo)**.** O fechamento transitivo G^T de G é o menor subconjunto de $V(G) \times V(G)$ que contém G e é transitivo.

DEFINIÇÃO 3.2.4 (Redução transitiva)**.** Dizemos que G^t é uma redução transitiva de G se as duas propriedades seguintes são válidas:

1. Há um caminho de u para v em G^t se, e somente se, há um caminho de u para v em G
2. Não há outro grafo com menos arcos que G^t que satisfaz a 1ª condição

DEFINIÇÃO 3.2.5 (Equivalência mínima)**.** Dizemos que G^e é uma equivalência mínima de G se as três propriedades seguintes são válidas:

- Há um caminho de u para v em G^e se, e somente se, há um caminho de u para v em G
- G^e é subgrafo de G
- Não há outro grafo com menos arcos que G^e que satisfaz a 1ª e 2ª condição

TEOREMA 6 (Aho et al. (1972))**.** *Para qualquer grafo direcionado e acíclico (DAG) G, há um único grafo G^t com a propriedade que $(G^t)^T = G^T$ e qualquer subgrafo próprio H de G^t satisfaz $H^T \neq G^T$. O grafo G^t é dado por:*

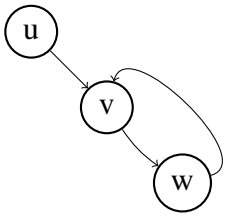
$$G^t = \bigcap_{G_i \in S(G)} G_i,$$

onde $S(G) = \{G_i | G_i^T = G^T\}$

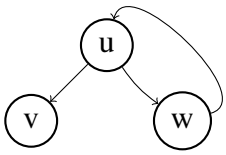
LEMA 7 (Aho et al. (1972))**.** *Seja G_1 e G_2 qualquer dois grafos DAGs (sobre o mesmo conjunto vértices) satisfazendo $G_1^T = {G_2}^T$. Se há um arco $\alpha \in G_1$ tal que $\alpha \notin G_2$, então $(G_1 - \alpha)^T = G_1^T = G_2^T$*

Demonstração. Definimos $\alpha = (u, v)$ como uma aresta redundante de G, se existe um caminho em G de u para v que não utiliza α, dessa maneira, α não acrescenta informação sobre a existência de um caminho em G. Logo, para demonstrar o Lema 1 devemos mostrar que α, como definido pela hipótese, é uma aresta redundante em G_1. Os seguintes passos, mostram como chegar a este resultado.

1. G_1 e G_2 são dois DAGs sobre o mesmo conjunto de vértices. (Pela hipótese)
2. $G_1^T = G_2^T$ (Pela hipótese)
3. Seja $\alpha = (u, v) \in G_1$ e $\alpha \notin G_2$ (Pela hipótese)
4. Existe uma caminho de u para v em G_2 que passa por outro vértice w. (Por 2 e 3)
5. w também é um vértice de G_1 (Por 1 e 4)
6. Existe um caminho de u para w e de w para v em G_1 (Por 2, 4, 5 e Definição de fecho transitivo)
7. Suponha (por absurdo) que α está no caminho de u para w



8. Suponha (por absurdo) que α está no caminho de w para v



   Em ambos os casos é gerada uma contradição, pois G_1 é acíclico.

9. Dessa maneira G_1 possui um caminho de u para v que não usa α, logo: $(G_1 - \alpha)^T = G_1^T = G_2^T$

□

LEMA 8 (Aho et al. (1972))**.** *Seja G um DAG então $S(G) = \{G_1 | G_1^T = G^T\}$ é fechado sobre a união e interseção.*

Demonstração. (União)

1. Seja G_1 e $G_2 \in S(G)$ (Pela hipótese)
2. $G_1^T = G_2^T = G^T$ (Por 1)

3. $G_1 \subseteq G_1^T = G^T$ (Definição de fecho transitivo)
4. $G_2 \subseteq G_2^T = G^T$ (Definição de fecho transitivo)
5. $(G_1 \cup G_2) \subseteq G^T$ (Por 3 e 4)
6. $(G_1 \cup G_2)^T \subseteq G^T$ (Pois G^T é transitivo)
7. $G_1 \subseteq G_1 \cup G_2$ (Definição de subconjunto)
8. $G^T = G_1^T \subseteq (G_1 \cup G_2)^T$ (Por 2)
9. $(G_1 \cup G_2)^T = G^T$ (Por 6 e 8)
10. $(G_1 \cup G_2) \in S(G)$ (Pela definição de $S(G)$)

(Interseção)

1. Seja G_1 e $G_2 \in S(G)$ (Pela hipótese)
2. $G_1^T = G_2^T = G^T$ (Por 1)
3. Seja $\{\alpha_1, \alpha_2, \ldots, \alpha_r\} = G_1 - (G_1 \cap G_2)$, a Figura 3.1 destaca que estas arestas estão estritamente G_1.

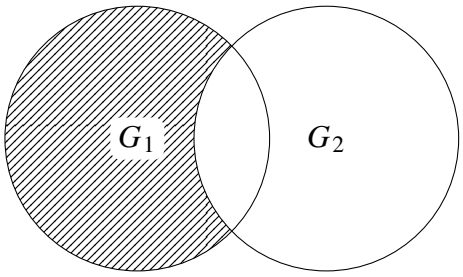
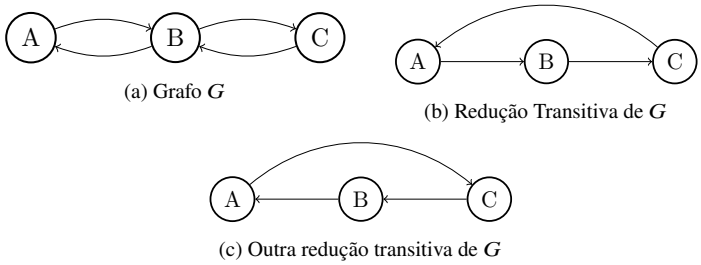


Figura 3.1: Dois grafos G_1 e G_2 representados através de um Diagrama de Venn, a parte estritamente em G_1 está destacada.

4. Seja $\alpha \in \{\alpha_1, \alpha_2, \ldots, \alpha_r\}$
5. $\alpha \in G_1$ (Por 3)
6. $\alpha \notin G_2$ (Por 3)
7. $(G_1 - \{\alpha_1\})^T = {G_1}^T$ (Por 2, 5, 6 e Lema 1)
8. $(G_1 - \{\alpha_1\} - \{\alpha_2\})^T = G_1^T$ (Por 2, 5, 6 e Lema 1)
9. ...
10. $(G_1 - \{\alpha_1\} - \{\alpha_2\} - \cdots - \{\alpha_r\})^T = G_1^T$ (Por 2, 5, 6 e Lema 1)



(a) Grafo G

(b) Redução Transitiva de G

(c) Outra redução transitiva de G

Figura 3.2: Problemas do caso cíclico

11. $(G_1 \cap G_2)^T = G_1^T = G^T$ (Por 10 e Definição de interseção)
12. $(G_1 \cap G_2) \in S(G)$ (Pela definição de $S(G)$)

□

3.2.1 Resultados do caso acíclico

O Teorema 6 nos dá dois resultados importantes sobre a redução transitiva, o primeiro é que se G é um grafo direcionado acíclico então sua redução transitiva é única, e segundo, a redução transitiva de G, G^t é um subgrafo de G obtido através da remoção de arestas redundantes.

3.2.2 Problemas para o caso com ciclos

Em um primeiro momento o caso cíclico é problemático, a Figura 3.2 mostra um grafo G cuja equivalência mínima é igual a ele mesmo, entretanto esta não é uma redução transitiva de G como mostra a Figura 3.2(b), ou seja, o conceito de equivalência mínima e redução transitiva se tornaram diferentes. Além disso, a redução transitiva mostrada na Figura 3.2(b) não é nem um subgrafo de G logo o algoritmo que vimos de ir removendo arestas redundantes do grafo original não funciona mais. Por fim, temos que a redução transitiva não é mais necessariamente única. Cada um desses problemas será resolvido nas seções subsequentes.

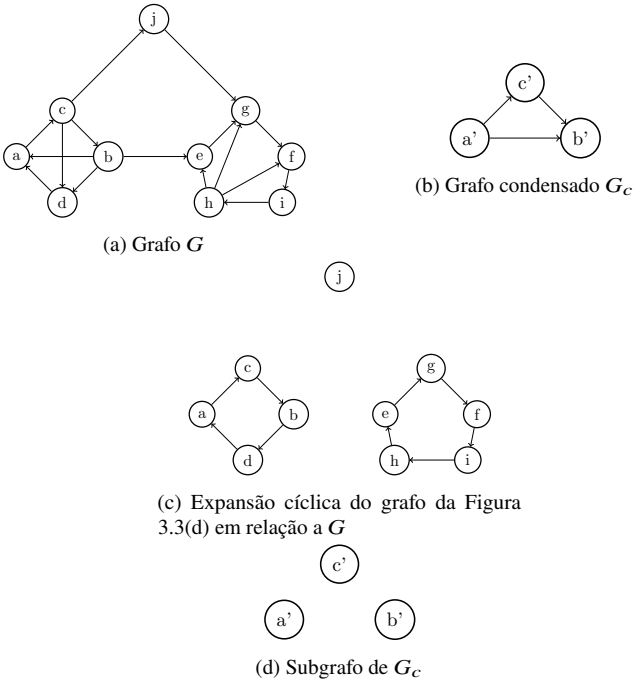
3.3 NOVAS DEFINIÇÕES

DEFINIÇÃO 3.3.1 (Equivalência de vértices)**.** Dois vértices u, v de um grafo direcionado G são chamados de equivalentes se ou $u = v$ ou G contém um ciclo que incide em ambos u e v.

DEFINIÇÃO 3.3.2 (Grafo acíclico equivalente (ou Grafo condensado))**.** Dado um grafo direcionado G, dizemos que G_c é o grafo condensado de G quando os vértices de G_c são as classes de equivalência sobre a equivalência de vértices, denotados por E_k, e os arcos de G_c satisfazem: $(E_i, E_j) \in G_c$, se, e somente se, existe um arco $(u, v) \in G$ tal que $u \in E_i$ e $v \in E_j$.

DEFINIÇÃO 3.3.3 (Expansão cíclica)**.** A operação de expansão cíclica, é a seguinte: Considere um grafo G e seu grafo condensado G_c, agora seja S um subgrafo de G_c, o grafo X é uma expansão cíclica de S em relação a G, se, e somente se:

1. $V(X) = V(G)$
2. $X_c = S$



(a) Grafo G

(b) Grafo condensado G_c

(c) Expansão cíclica do grafo da Figura 3.3(d) em relação a G

(d) Subgrafo de G_c

Figura 3.3: Exemplo de expansão cíclica

3. Para cada E_i de X_c não unitário, X contém um ciclo simples, incidente a todos os vértices de E_i que estão em X.

4. Para cada arco $(E_i, E_j) \in S$, $E_i \neq E_j$, há exatamente um arco $(u, v) \in C$, que satisfaz $u \in E_i$ e $v \in E_j$.

   A Figura 3.3 mostra um exemplo de uso.

TEOREMA 9 (Aho et al. (1972))**.** *Dado um grafo direcionado G, seja G_c seu grafo condensado e G_c^t a redução transitiva única de G_c dada pelo Teorema 6. Então o grafo direcionado H satisfaz $H^T = G^T$ e tem o menor número de arcos que qualquer outro grafo se, e somente se, H é uma expansão cíclica de G_c^t*

LEMA 10 (Aho et al. (1972))**.** *Se $H^T = G^T$, então $H_c^T = G_c^T$, onde H_c e G_c são os grafos condensados de H e G, respectivamente.*

Demonstração. Tomando a hipótese como verdadeira, mostraremos que H_c^T tem o mesmo conjunto de vértices e arestas de G_c^T.

1. $H^T = G^T$ (Pela hipótese)

2. Existe um caminho de u para v em $H \equiv$ Existe um caminho de u para v em G (Definição de fecho transitivo)

3. Classes de equivalência de vértices de G é igual a H (Por 1 e 2)

4. $V(G_c) = V(H_c)$ (Por 3)

5. (Por contradição) Assuma que existe um arco $(E_i, E_j) \in H_c^T$ e $(E_i, E_j) \notin G_c^T$

6. Dessa maneira, existe u, v onde $u \in E_i$ e $v \in E_j$ tal que existe um caminho de u para v em H mas não em G. Contradição! (Por 2)

7. O mesmo argumento pode ser usado para um arco $(E_i, E_j) \in G_c^T$ e $(E_i, E_j) \notin H_c^T$
8. $\alpha \in H_c^T \equiv \alpha \in G_c^T$
9. $H_c^T = G_c^T$ (Por 4 e 8)

□

LEMA 11 (Aho et al. (1972))**.** *Se H tem um grafo condensado H_c satisfazendo $H_c^T = G_c^T$, então H é ou uma expansão cíclica de G_c^t ou H contém mais arcos que qualquer expansão cíclica de G_c^t.*

Demonstração. 1. $H_c^T = G_c^T$ (Pela hipótese)

2. $G_c^t \subseteq H_c$ (Pelo Teorema 6)
3. $(E_i, E_j) \in H_c \Rightarrow \exists u \in E_i$ e $v \in E_j$ tal que $(u, v) \in H$ (Definição de grafo condensado)
4. Além dos arcos de 3, H precisa ter pelo menos um ciclo simples para cada $E_i \in H_c$ onde E_i não é unitário.
5. Para minimizar 3 e 4, H deve ser uma expansão cíclica de G^t
6. Logo, ou H é uma expansão cíclica de G^t ou tem mais arcos que qualquer expansão cíclica de G^t

□

LEMA 12 (Aho et al. (1972))**.** *Toda expansão cíclica de G_c^t tem o mesmo número de arcos.*

Demonstração.
$$\underbrace{|(E_i, E_j) \in G_c^t|}_{\text{Arcos entre classes de equivalência}} + \underbrace{\sum_{E_i \in G_c^t} (|E_i| \cdot [|E_i| > 1])}_{\text{Tamanhos dos ciclos}} - \underbrace{\sum_{E_i \in G_c^t} [|E_i| = 1]}_{\text{Remoção de loops}}$$

□

LEMA 13 (Aho et al. (1972))**.** *Se H é uma expansão cíclica de G_c^t, então $H^T = G^T$*

Demonstração. O objetivo é mostrar que $(u, v) \in H^T \Leftrightarrow (u, v) \in G^T$. Para isso mostraremos que $(u, v) \in G^T \Rightarrow (u, v) \in H^T$ (ida) e $(u, v) \notin G^T \Rightarrow (u, v) \notin H^T$ (volta)
($\Rightarrow$)

1. $(u, v) \in G^T$ (Pela hipótese)
2. G contém um caminho de u para v (Definição de fecho transitivo)
3. Caso (i) u e v pertencem a mesma classe de equivalência
   - Logo, existe um caminho de u para v em H, pois u e v estarão em um mesmo ciclo simples.
   - $(u, v) \in H^T$
4. Caso (ii) u e v estão em classes de equivalência diferentes.
   - $u \in E_i$
   - $v \in E_j$

- Existe um caminho de E_i para E_j em G_c
- Existe um caminho de E_i para E_j em G_c^t
- Existe um caminho de u para v.
- $(u, v) \in H^T$

($\Leftarrow$)

1. $(u, v) \notin G^T$ (Pela hipótese)
2. G não contém um caminho de u para v
3. u e v pertencem a classes de equivalência diferentes
4. $u \in E_i$
5. $v \in E_j$
6. Não existe caminho de E_i para E_j em G_c
7. Não existe caminho de E_i para E_j em G_c^t
8. Não existe caminho de u para v em H
9. $(u, v) \notin H^T$

□

3.4 COMPLEXIDADE DE COMPUTAR A REDUÇÃO TRANSITIVA

3.4.1 Matriz de adjacência

A *matriz de adjacência* de um grafo direcionado G com n vértices, é uma matriz $n \times n$, M, indexada pelos vértices de G dada por:

$$M[u, v] = \begin{cases} 1, & \text{se } (u, v) \in G. \\ 0, & \text{se } (u, v) \notin G. \end{cases}$$

Como visto em (Aho et al., 1972) o tempo para computar o fecho transitivo é assintoticamente equivalente ao tempo de computar a multiplicação de matrizes booleanas, além de ser polinomial, dessa maneira temos o seguinte teorema.

TEOREMA 14 (Aho et al. (1972))**.** *Se há um algoritmo para computar o fecho transitivo de um grafo com n vértices em tempo $O(n^\alpha)$, então há um algoritmo para computar a redução transitiva em tempo $O(n^\alpha)$.*

Demonstração. Podemos computar a redução transitiva de um grafo G com n vértices como segue:

1. Encontre G_1, o grafo condensado de G
2. Seja G_2 o grafo formado por G_1 removendo loops.
3. Seja M_1 a matriz de adjacência de G_2, e seja M_2 a matriz de adjacência de G_2^T

4. Compute $M_3 = M_1 M_2$, e seja G_3 o grafo cuja matriz de adjacência é M_3.

5. Então $G_1^t = G_1 - G_3$

6. Seja G^t uma expansão cíclica de de G_1^t

Complexidade: 1, 2, 5 e 6 é $O(n^2)$, e 3, 4 é $O(n^\alpha)$

Corretude: Mostrar que $G_3 \coloneqq \{$é o conjunto dos arcos redundantes$\}$

$G_3 \coloneqq \{(u, v) \quad | \quad \exists w$ tal que existe um caminho de u para w e de w para v em G_1 que não utiliza $(u, v)\}$ $\}$

$$M_1 = \mathbf{A} = \begin{bmatrix} 0 & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & 0 & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & 0 \end{bmatrix} \qquad a_{ij} = \begin{cases} 1, & \text{se } (i,j) \in G_1 \text{ e } i \neq j \\ 0, & \text{caso contrário.} \end{cases}$$

$$M_2 = \mathbf{B} = \begin{bmatrix} 0 & b_{1,2} & \cdots & b_{1,n} \\ b_{2,1} & 0 & \cdots & b_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n,1} & b_{n,2} & \cdots & 0 \end{bmatrix} \qquad b_{ij} = \begin{cases} 1, & \text{se existe um caminho} \\ & \text{de } i \text{ para } j \text{ em } G_2 \\ 0, & \text{caso contrário.} \end{cases}$$

$$M_3 = \mathbf{C} = \mathbf{A} \times \mathbf{B} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ c_{n,1} & c_{n,2} & \cdots & c_{n,n} \end{bmatrix}$$

$c_{i,j} = a_{i,1}b_{1,j} \vee \cdots \vee a_{i,j}b_{j,j} \vee \cdots \vee a_{i,n}b_{n,j}$

$c_{i,j} = a_{i,1}b_{1,j} \vee \cdots \vee a_{i,j}0 \vee \cdots \vee a_{in}b_{nj}$

$c_{i,j} = a_{i,1}b_{1,j} \vee \cdots \vee a_{i,j-1} \vee a_{i,j+1} \vee \cdots \vee a_{i,n}b_{n,j}$

$c_{i,j} = 1$ se existe um w tal que $a_{i,w} = 1$ e $b_{w,j} = 1$, onde $w \neq j$

$a_{i,w}$ não possui (i, j), pois como $w \neq j$ e $a_{i,w}$ é um caminho com apenas uma aresta.

Para o caminho de w para j, este também não pode utilizar (i, j) pois dessa maneira teríamos um ciclo, que contradiz o fato de G_1 ser acíclico.

Ou seja, $c_{i,j} = 1$ se existe um w tal que existe um caminho de i para w e de w para j em G_1 que não utiliza (i, j)

□

4 PROBLEMAS

4.1 INTRODUÇÃO

Neste capítulo serão vistas aplicações dos teoremas que foram estudados nos capítulos anteriores, em problemas relacionados a programação competitiva. Cada problema acompanha seu enunciado, que descreve o problema, seus limites, que indicam, por exemplo, tamanho da entrada e duração máxima do tempo de execução do programa, e em seguida, são dados exemplos de entrada, que buscam deixar mais claros os detalhes do enunciado, e por fim, uma resolução que utiliza os conceitos já estudados. Os limites de tempo que acompanham cada problema consideram que uma máquina convencional em 2023 consegue executar cerca de 10^8 operações elementares por segundo seguindo o modelo de memória RAM.

4.2 PROBLEMA 1: ANIVERSÁRIO

Enunciado: Hoje é aniversário da pequena Dasha. Nesta ocasião, cada um de seus n amigos lhe dará um cartão com uma mensagem escrita nele, fato é que todas as mensagens serão diferentes. Dasha recolheu todos esses presentes e decidiu jogar fora alguns deles de maneira que os cartões restantes no conjunto sejam *estilosos*. A aniversariante considera um conjunto de cartões *estilosos* se para qualquer dois cartões distintos nesse conjunto a mensagem escrita em um deles não é substring da outra. Uma substring de uma string s é um segmento contínuo de s. São dadas as n mensagens, ajude Dasha a encontrar o conjunto *estiloso* máximo. (Adaptado de: Pyaderkin e Andreev (2015))

Limites: $1 \leq n \leq 750$, tamanho da entrada $\leq 10^7$ caracteres, as mensagens são compostas apenas dos caracteres “A” e “B”, e o tempo limite = 3 segundos.

Exemplo de entrada:

Figura 4.1: Exemplo de cartões de aniversários recebidos

Para essa entrada, Dasha poderia escolher os seguintes cartões: AABAB e ABABB, por exemplo, resultando em um conjunto *estiloso* de tamanho máximo.

Resolução: A ideia é criar um conjunto ordenado **P** utilizando como conjunto base o conjunto das mensagens combinada com a relação de substring, pois esta é uma relação de ordem. A partir dessa formalização, é possível notar que a definição de conjunto *estiloso* definido no problema, se reduz a encontrar a maior anticadeia de **P**, que como visto na demonstração do Teorema de Dilworth no Capítulo 1, pode ser resolvido com uma redução ao problema de encontrar o emparelhamento máximo em um grafo bipartido.

Portanto, precisamos resolver dois subproblemas: (1) montar o grafo subjacente de **P** utilizando as strings dadas pela entrada e (2) encontrar o emparelhamento máximo em um grafo bipartido. O subproblema (2), dados os limites do problema, pode ser resolvido com Algoritmo de Kuhn com complexidade de tempo $O(n^3)$, visto que o grafo subjacente de **P** pode ser denso. Para (1) como as strings podem ser grandes ($\leq 10^7$) é necessário um maior cuidado, para isso podemos aproveitar da estrutura Autômato de Aho-Corasick das strings dadas na entrada.

4.3 PROBLEMA 2: BONECAS RUSSAS

Enunciado: Dilworth é o colecionador mais proeminente do mundo de bonecas russas: ele literalmente tem milhares delas! Você sabe, as bonecas ocas de madeira de tamanhos diferentes, das quais a menor boneca está contida na segunda menor, e esta boneca por sua vez está contida na próxima e assim por diante. Um dia ele se pergunta se há outra maneira de aninhar as bonecas para acabar com menos bonecas aninhadas? Afinal, isso tornaria sua coleção ainda mais magnífica! Ele desembala cada boneca aninhada e mede a largura e a altura de cada boneca contida. Uma boneca com largura w_1 e altura h_1 caberá em outra boneca de largura w_2 e altura h_2 se e somente se $w_1 < w_2$ e $h_1 < h_2$. Você pode ajudá-lo a calcular o menor número possível de bonecas aninhados para montar a partir de sua enorme lista de medidas? (Adaptado de Björklund (2007))

Limites: Tamanho da lista $\leq 2 \times 10^4$ e o tempo limite = 1 segundo.

Exemplo de entrada:

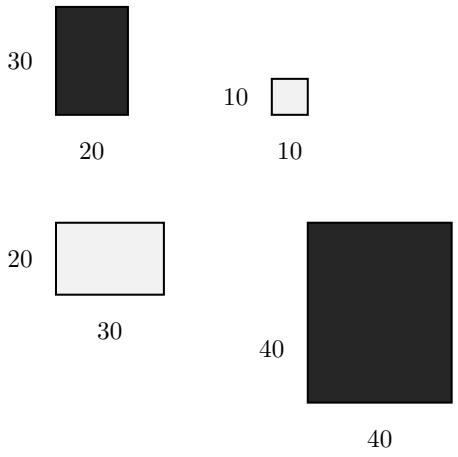


Figura 4.2: Exemplo de bonecas russas representadas apenas por suas medidas

Para essa entrada, Dilworth pode aninhar cada uma das bonecas coloridas de preto na Figura 4.2 com exatamente uma outra boneca colorida de cinza claro, isso resulta em um número mínimo de conjuntos de bonecas, essa solução é ilustrada na Figura 4.3.

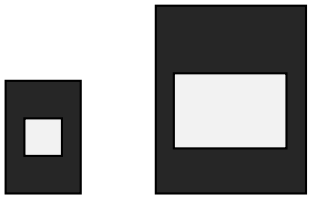


Figura 4.3: Solução para a entrada exemplo do problema das bonecas russas

Resolução: Similar ao problema anterior podemos criar um poset $\mathbf{P} = (P, \leq)$, usando os retângulos que representam as bonecas como o conjunto P, e a relação de "uma boneca estar dentro da outra" como definida pelo problema, e do mesmo modo, o problema de encontrar a menor partição de bonecas aninhadas é equivalente ao problema de encontrar a maior anticadeia de $\mathbf{P}$. Entretanto, devido ao tamanho do problema ($n \leq 10^4$), não é possível utilizar o algoritmo de Kuhn para encontrar a maior anticadeia dentro do tempo limite (1 segundo), pois assim como no caso anterior, como grafo subjacente de $\mathbf{P}$ é denso, a complexidade do algoritmo de Kuhn seria $O(n^3)$, cerca de 10^{12} operações elementares. Desse modo, um caminho para encontrar a

solução é encontrar qual o formato de uma anticadeia desse problema, para isso, observe a Figura 4.4, nela o Diagrama de Hasse é feito utilizando x, y, como a largura e altura de uma boneca, respectivamente. Agora, considere uma enumeração da esquerda para direita, de cima para baixo, a partir de 1 de cada uma das bonecas representadas por pontos na Figura 4.4, e projete no eixo x essa enumeração. A partir dessa projeção temos que qualquer anticadeia de **P** corresponde uma subsequência crescente dessa projeção, e portanto podemos resolver o problema original, encontrando a maior subsequência crescente (*LIS*) desta projeção com complexidade de tempo $O(n^2)$ por exemplo.

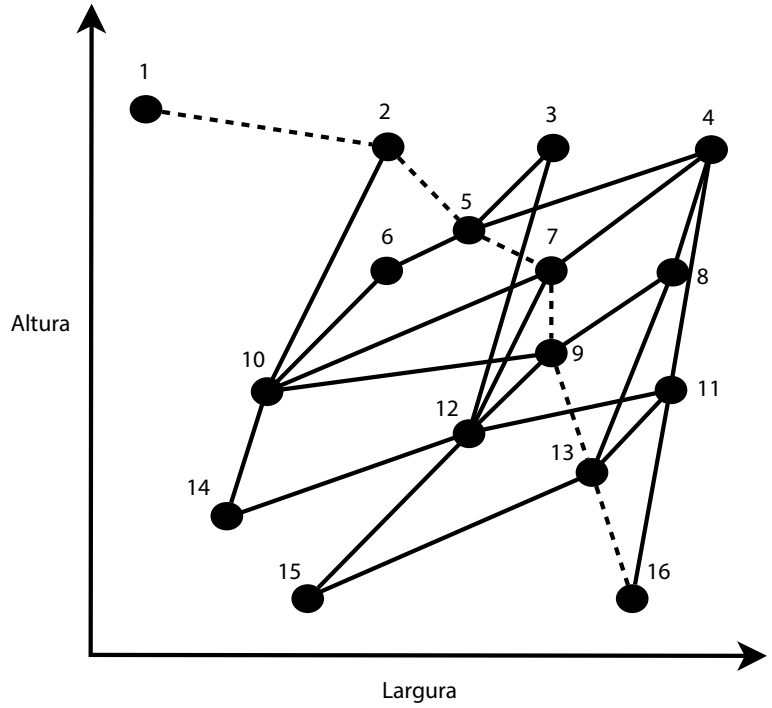


Figura 4.4: "Diagrama de Hasse" de uma instância do problema Bonecas Russas. A sua projeção no eixo x é igual a 1, 14, 10, 15, 2, 6, 5, 12, 3, 7, 9, 13, 16, 8, 11, 4 (Adaptado de Björklund (2007))

4.4 PROBLEMA 3: ORGANIZANDO CARTAZES

Enunciado: Um quadro de avisos em sua universidade está cheio de cartazes que anunciam diversos eventos. Uma nova política foi estabelecida de que dois cartazes não podem se sobrepor. Você foi contratado para remover alguns cartazes de tal maneira que os cartazes restantes sigam a nova política. Para não prejudicar tanto os estudantes, você deve remover o número mínimo de cartazes para atingir esse objetivo. Você não pode remover um cartaz e colocá-lo em outra posição; todos os cartazes podem ou ser removidos ou devem permanecer em suas posições originais. Cada cartaz é um retângulo paralelo ao eixo x, cuja intersecção com outro cartaz nunca é igual a ele mesmo. (Adaptado de Codechef (2010))

Limites: Quantidade de cartazes $\leq 10^2$ e o tempo limite = 0.6 segundo.

Exemplo de entrada:

Para a entrada da Figura 4.5 podemos remover o cartaz mais largo, como os cartazes restantes não se intersectam, o resultado segue a nova política.

Resolução: Seguindo o passo a passo dos problemas anteriores, o processo para resolver esse problema seria (1) criar um *poset* $\mathbf{P} = (P, \leq)$ onde P é o conjunto dos cartazes e (2) utilizar a relação de "sobreposição" definida no enunciado como uma relação de ordem. Porém, neste

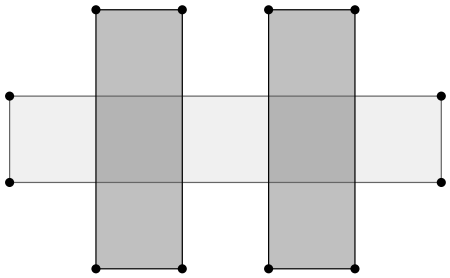


Figura 4.5: Três cartazes dispostos no plano, onde um cartaz maior intersecta outros dois menores, enquanto estes não se intersectam entre si.

caso, a relação de "sobreposição" exatamente como descrita não é uma relação de ordem, pois não é transitiva. Para garantirmos a transitividade podemos notar que o problema que faz com que a relação de "sobreposição" não seja transitiva está relacionado com o fato de não estar definido quem está sobre quem, para isso podemos arbitrariamente definir que o retângulo mais largo é aquele que está abaixo, e isso garante que a relação de "sobreposição" seja uma ordem estrita, e desse modo, podemos proceder como antes.

4.5 PROBLEMA 4: GENTRIFICAÇÃO

Enunciado: Os moradores de Townsville deixaram claro para a prefeita que estão muito preocupados com a gentrificação, um processo pelo qual pessoas ricas se mudam para uma cidade em grande número, deslocando as pessoas que atualmente moram lá. A prefeita de Townsville sabe uma ou duas coisas sobre isso e gostaria de tranquilizar as pessoas determinando o pior cenário possível.

Townsville é composta por N bairros, com M estradas de mão única passando entre elas. A i-ésima estrada vai do bairro A_i ao bairro B_i. Uma multidão de migrantes ricos se mudará para a cidade de uma só vez, gentrificando imediatamente qualquer bairro para o qual decidem se mudar.

A prefeita conhece os seguintes fatos sobre esses novos moradores abastados: (1) eles gostam de visitar outros bairros gentrificados. Se houver uma maneira de ir de seu bairro natal para outro bairro gentrificado, eles com certeza irão visitá-lo, (2) eles nunca caminham a lugar nenhum, eles só dirigem. Consequentemente, eles ficarão muito zangados se acabarem em algum bairro sem poder dirigir de volta para casa.

Juntando esses fatos, isso significa que os migrantes ricos só irão gentrificar dois bairros u e v, se houver uma sequência de estradas conectando u a v e que conectem v para u.

Dada essas restrições e o layout das estradas em Townsville, qual é o número máximo de bairros que podem ser gentrificados? (Adaptado de Taravilse (2015))

Limites: $1 \leq N \leq 5 \times 10^2, 1 \leq M \leq 2 \times 10^4$

Exemplos de entrada:

A Figura 4.6 mostra exemplos de entrada para o problema, representando bairros como vértices e estradas como arcos. Cada instância tem uma solução destacada. Note que em 4.6(b) não é possível escolher mais de um vértice pois para qualquer dois vértices distintos u, v haveria uma caminho de u para v mas não haveria um caminho de v para u. Por outro lado, na Figura

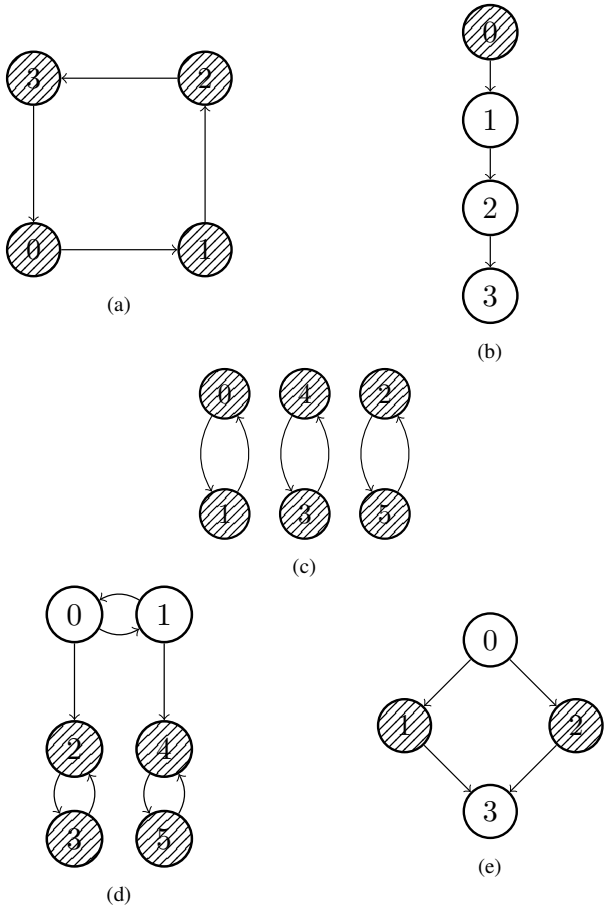


Figura 4.6: Exemplos de entrada para o problema Gentrificação

4.6(e) é possível escolher os vértices 1 e 2, pois as condições (1) e (2) resumidas em: "se há um caminho de u para v então é necessário ter um caminho de v para u", é verdadeira por vacuidade.

Resolução: Inicialmente, considere o grafo direcionado G definido utilizando os bairros como vértices e as estradas como arcos. Dois vértices u e v podem ser gentrificados ou se ambos se alcançam ou se nenhum dos dois se alcança. Isso é mesmo que dizer que ou u e v estão no mesmo componente fortemente conexo, ou u e v estão em componentes fortemente conexos que não são alcançáveis entre si e desse modo, podemos considerar que, ou todos os bairros em um mesmo componente conexo serão gentrificados ou nenhum dos bairros representados será gentrificado.

A partir dessa formalização, podemos construir um *poset* $\mathbf{P} = (P, \leq)$ com uma função de pesos w, onde P são os componentes fortemente conexos de G, e para $x, y \in P, x \leq y$ se, e somente se, x alcança y, e por fim $w(x) :=$ quantidade de vértices de G que foram condensados em x. Agora, o problema de encontrar a quantidade máxima de bairros que pode ser gentrificada se reduz ao problema de encontrar uma anticadeia de maior soma de pesos, que como vimos no Capítulo 1, pode ser resolvido utilizando o Teorema de Dilworth (Generalizado).

5 CONCLUSÃO

Neste trabalho o conceito de ordem foi apresentado através de duas aplicações principais: (1) representação similar ao conceito de Diagrama de Hasse para grafos direcionados, e (2) aplicações do Teorema de Dilworth no contexto da programação competitiva. Em resumo, no Capítulo 1, foram apresentadas as demonstrações dos teoremas de Mirsky e Dilworth clássicas, assim como suas versões generalizadas para *posets* com pesos. No Capítulo 2, foi mostrada uma maneira de computar a redução transitiva de um grafo direcionado qualquer, em tempo polinomial, e por fim, no Capítulo 3, foram apresentadas aplicações do Teorema de Dilworth em problemas da programação competitiva, esses exemplos mostraram maneiras de trabalhar com uma ordem em problemas de brinquedo. Uma implementação para o algoritmo de redução transitiva e soluções para os problemas visto pode ser encontrada em "Soluções de problemas relacionados a conjuntos ordenados"[1].

[1]Disponível em: https://github.com/pedrov3/solucoes-conjuntos-ordenados

REFERÊNCIAS

Aho, A. V., Garey, M. R. e Ullman, J. D. (1972). The transitive reduction of a directed graph. https://www.cs.tufts.edu/~nr/cs257/archive/al-aho/transitive-reduction.pdf. Acessado em 25/06/2023.

Björklund, A. (2007). MDOLLS - nested dolls. https://www.spoj.com/problems/MDOLLS/. Acessado em 25/06/2023.

Bondy, J. A. (2008). *Graph Theory with Applications*. Springer Publishing Company.

Codechef (2010). Codechef - problem posters. https://www.codechef.com/problems/POSTERS/. Acessado em 25/06/2023.

Davey, B. A. e Priestley, H. A. (2002). *Introduction to Lattices and Order*, páginas 1–32. Cambridge University Press.

Dilworth, R. P. (1950). A decomposition theorem for partially ordered sets. *Annals of Mathematics*, 51(1):161–166.

Felsner, S. (2020). Lecture 02 - october 9, 2020 (introduction to order theory). https://www.youtube.com/watch?v=sbKDJbD379E&list=PL5rqYzyihIQ0nzfnsEKxxedCpbNQoifgg&ab_channel=Order%26Geometry. Acessado em 25/06/2023.

Goemans, M. X. (2004). Topics in combinatorial optimization - lecture 6. https://ocw.mit.edu/courses/18-997-topics-in-combinatorial-optimization-spring-2004/72c5d7935dea64f0e2773ad376f83d5d_co_lec6.pdf. Acessado em 25/06/2023.

Might, M. (2012). Partial orders. https://matt.might.net/articles/partial-orders/?s=09. Acessado em 25/06/2023.

Mirsky, L. (1971). A dual of dilworth's decomposition theorem. *The American Mathematical Monthly*, 78(8):876–877.

Pyaderkin, M. e Andreev, R. (2015). Codeforces - contest 590, problem e - birthday. https://codeforces.com/contest/590/problem/E. Acessado em 25/06/2023.

SBC (2013). Problem 1468 - beecrowd. https://www.beecrowd.com.br/judge/en/problems/view/1468. Acessado em 25/06/2023.

Taravilse, L. (2015). Problem C: Gentrification - facebook hacker cup 2015, round 3. https://www.facebook.com/codingcompetitions/hacker-cup/2015/round-3/problems/C. Acessado em 25/06/2023.

Trotter, W. T. (2015). Lecture 14 - math 3012 open resources. https://sites.gatech.edu/math3012openresources/lecture-videos/lecture-14/. Acessado em 25/06/2023.