Parameterisierte Algorithmen

WS 2009/10 in Trier

Henning Fernau
fernau@uni-trier.de

Parameterisierte Algorithmen Gesamtübersicht

- Einführung
- Grundbegriffe
- Problemkerne
- Suchbäume
- Graphparameter
- Weitere Methoden
- Komplexitätstheorie—parameterisiert

Bisher ... heute

Suchbäume erlauben (oft verhältnismäßig) effiziente parameterisierte Alg.

Graphprobleme werden oft einfacher, wenn sie auf gewisse Graphklassen eingeschränkt werden.

Das letzte Mal haben wir dazu Kantengraphen betrachtet.

Heute werden wir uns planare Graphen ansehen.

Eulers Formel: PIS

PLANARE UNABHÄNGIGE MENGE (PIS)
Eingabe: ein planarer Graph $G = (V, E)$
Parameter: eine natürliche Zahl k
Frage: Gibt es eine unabhängige Menge $I \subseteq V$ mit $|I| = k$?

Hinweis: IS W[1]-hart auf allgemeinen Graphen, polynomiell auf Kantengraphen.

Satz 1 *Durch triviales Verzweigen bei kleingradigen Knoten:* $\mathcal{O}^*(6^k)$.

Algorithm 1 A simple search tree algorithm for PIS, called PIS-ST-simple

Input(s): planar graph $G = (V, E)$, positive integer k
Output(s): YES if G has an independent set I with $|I| = k$; NO otherwise

```
if k ≤ 0 then
  return YES
else if V = ∅ then
  return NO
else
  Let v be a vertex of lowest degree in G.
  {One vertex from N[v] will be in any maximal independent set.}
  for all x ∈ N[v] do
    if PIS-ST-simple(G[V \ N[x]], k − 1) then
      return YES
  {All trials have failed.}
  return NO
```

Mehr Mathematik: PIS

Satz 2 *(Borodin et al.) Jeder zusammenhängende planar eingebettete Graph mit wenigstens zwei Knoten besitzt*

1. *entweder zwei Knoten mit Gradsumme höchstens fünf*
2. *oder zwei Knoten vom Abstand höchstens zwei mit Gradsumme höchstens sieben*
3. *oder ein Dreieck mit zwei inzidenten Knoten mit Gradsumme höchstens neun*
4. *oder zwei über eine Kante* $\{u, v\}$ *benachbarte Dreiecke mit Gradsumme von* u *und* v *höchstens elf.*

Hinweis Damit: $\mathcal{O}^*(5.17^k)$.

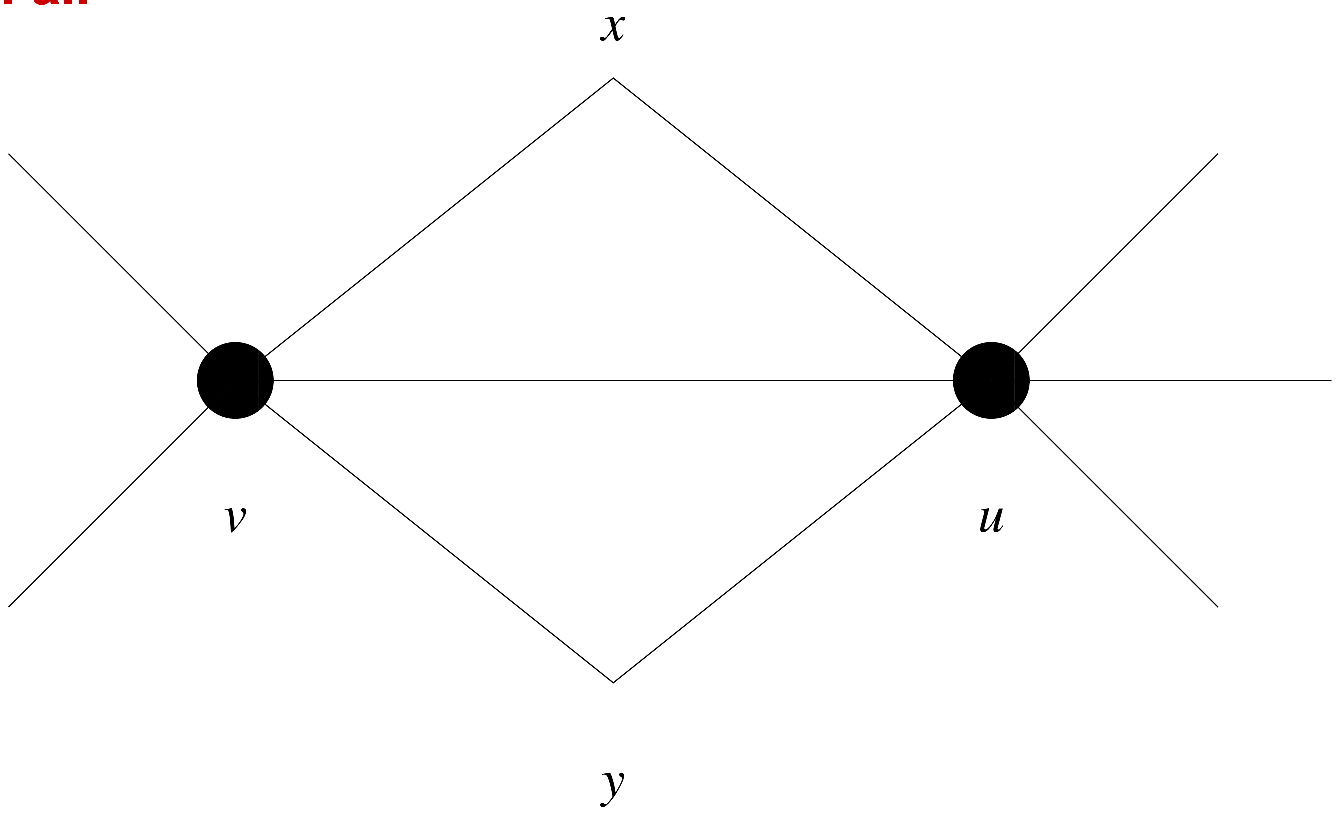
Algorithm 2 A more advanced search tree algorithm for PIS, called PIS-ST

Input(s): planar graph $G = (V, E)$, positive integer k

Output(s): YES if G has an independent set I with $|I| = k$; NO otherwise

- **if** $k \leq 0$ **then**
  - return YES
- **else if** $V = \emptyset$ **then**
  - return NO
- **else**
  - Let v be a vertex of lowest degree in G.
  - {One vertex from $N[v]$ will be in any maximal independent set.}
  - **if** $\deg(v) \leq 4$ **then**
    - **for all** $x \in N[v]$ **do**
      - **if** PIS-ST($G[V \setminus N[x]], k - 1$) **then**
        - return YES
    - return NO
  - **else**
    - {The fourth case of Theorem 2 applies. Details to follow...}

Der vierte Fall



$\{N(v) = \{x, y, u, v_1, v_2\}$ with $N(u) = \{x, y, v, u_1, u_2, u_3\}\}$

if PIS-ST($G[V \setminus N[v]], k-1$) **then**

  return YES; {Branch 1: v in IS ?}

else if PIS-ST($G[V \setminus N[u]], k-1$) **then**

  return YES; {Branch 2: v is not in IS, but u ?}

else if PIS-ST($G[V \setminus N[x]], k-1$) **then**

  return YES; {Branch 3: v, u is not in IS, but x ?}

else if PIS-ST($G[V \setminus N[y]], k-1$) **then**

  return YES; {Branch 4: v, u, x is not in IS, but y ?}

else

  {Now v, u, x, y is not in IS, so one of the v_i and one of the u_j must be in a maximal independent set.}

  for all $i = 1, 2; j = 1, 2, 3$ **do**

    if PIS-ST($G[V \setminus (N[v_i] \cup N[u_j])], k-2$) **then**

      return YES

  return NO

Zur Algorithmenanalyse

Korrektheit: Folgt mit dem Satz von Borodin etc.

Laufzeitabschätzung: T erfüllt

$$T(k) \leq 4T(k-1) + 6T(k-2).$$

Das liefert die Behauptung:

$$c^k \approx 4c^{k-1} + 6^{k-2}$$

$$c^2 - 4c + 6 = 0$$

$\rightsquigarrow c = 2 + \sqrt{10} \leq 5.17.$

Ein komplizierteres Beispiel: PDS

PLANARE DOMINIERENDE MENGE (PDS)
Eingabe: ein planarer Graph $G = (V, E)$
Parameter: eine natürliche Zahl k
Frage: Gibt es eine dominierende Menge $D \subseteq V$ mit $|D| \leq k$?

Hinweis: Problemkern von $67k$ (sehr kompliziert)

Problem beim Verzweigen: es entstehen zwei Arten von Knoten:
—Solche, die noch abgedeckt werden müssen und
—solche, die man schon dominiert hat, die aber noch in die Dominierungsmenge aufgenommen werden sollten.

Eine Verallgemeinerung durch Annotation

PLANARE ANNOTIERTE DOMINIERENDE MENGE (PADS)
Eingabe: ein planarer Graph $G = (V, E)$ und Knotenmenge B
Parameter: eine natürliche Zahl k
Frage: Gibt es eine Menge $D \subseteq V$, $|D| \leq k$, die jeden Knoten aus B dominiert ?

Wir nennen die Knoten aus B schwarz (black) und deren Komplement W (weiß).
M.a.W.: Wir betrachten planare Schwarz-Weiß-Graphen.

Frage: Gibt es stets kleingradigen schwarzen Knoten ?
“Ohne Weiteres” nicht: Betrachte Sterngraph, bei dem nur das Zentrum schwarz.

Problemlösung Reduktionsregeln

R1: Lösche Kanten zwischen weißen Knoten.

R2: Lösche weiße Knoten vom Grad 1.

R3: Sei u ein weißer Knoten vom Grad 2 mit schwarzen Nachbarn v_1 und v_2.

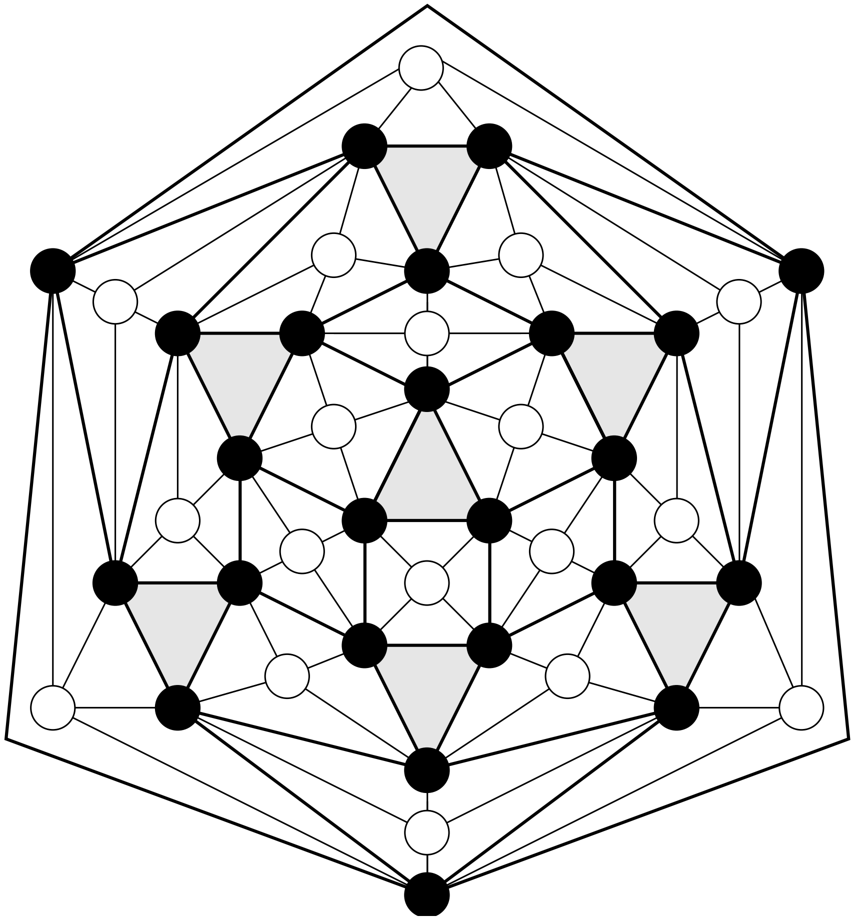
R3(a): Gibt es Kante zwischen v_1 und v_2, so lösche u.

R3(b): Gibt es einen weiteren Knoten $v_3 \in N(v_1) \cap N(v_2)$, so lösche u.

R4: Sei u ein weißer Knoten vom Grad 3 mit schwarzen Nachbarn v_1, v_2 und v_3. Gibt es Kanten $\{v_1, v_2\}$ und $\{v_2, v_3\}$, so lösche u.

Lemma 3 *Die Reduktionsregeln sind korrekt.*

Satz 4 *In einem reduzierten planaren Schwarz-Weiß-Graphen gibt es einen schwarzen Knoten vom Maximalgrad sieben.*

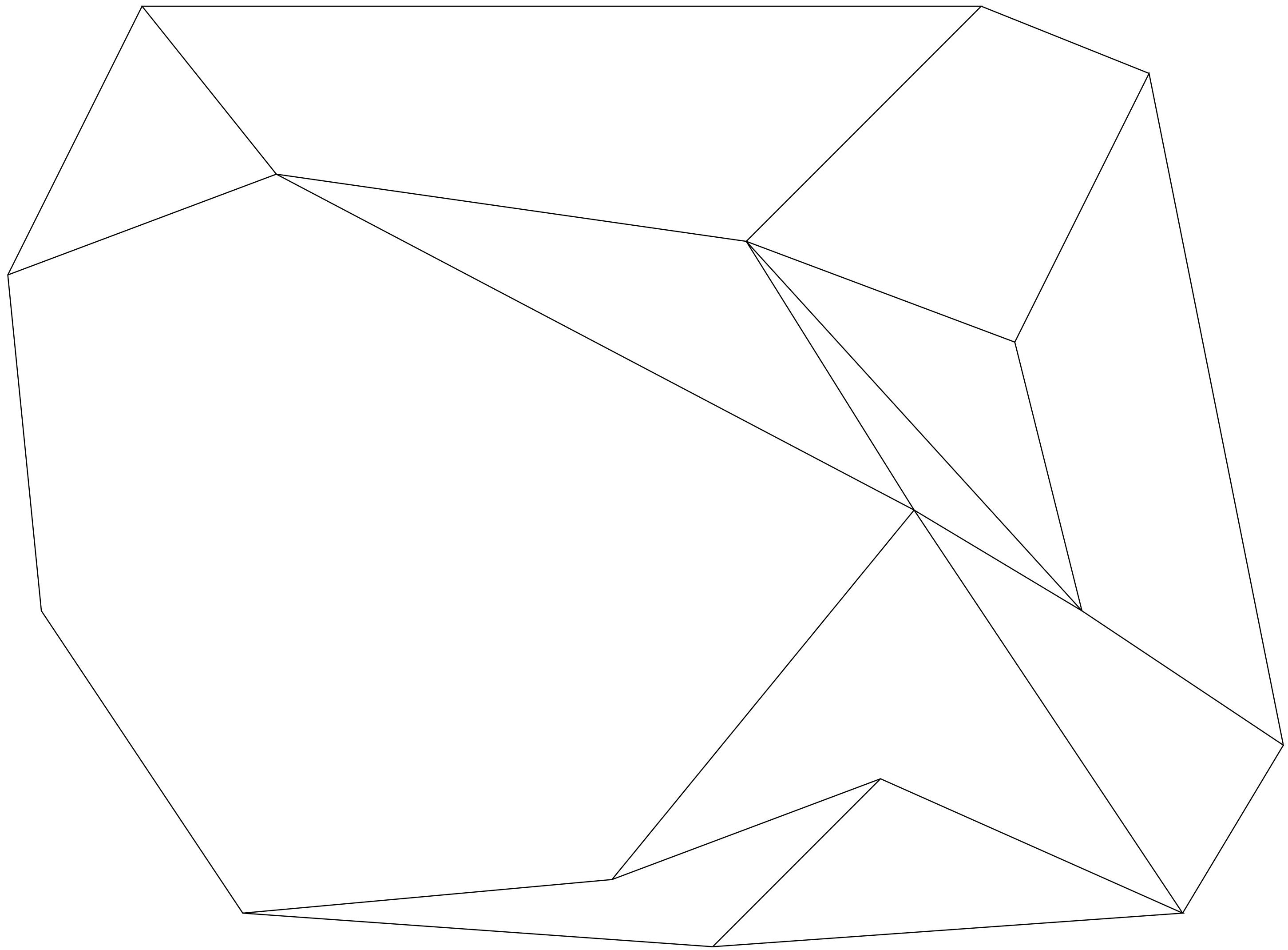


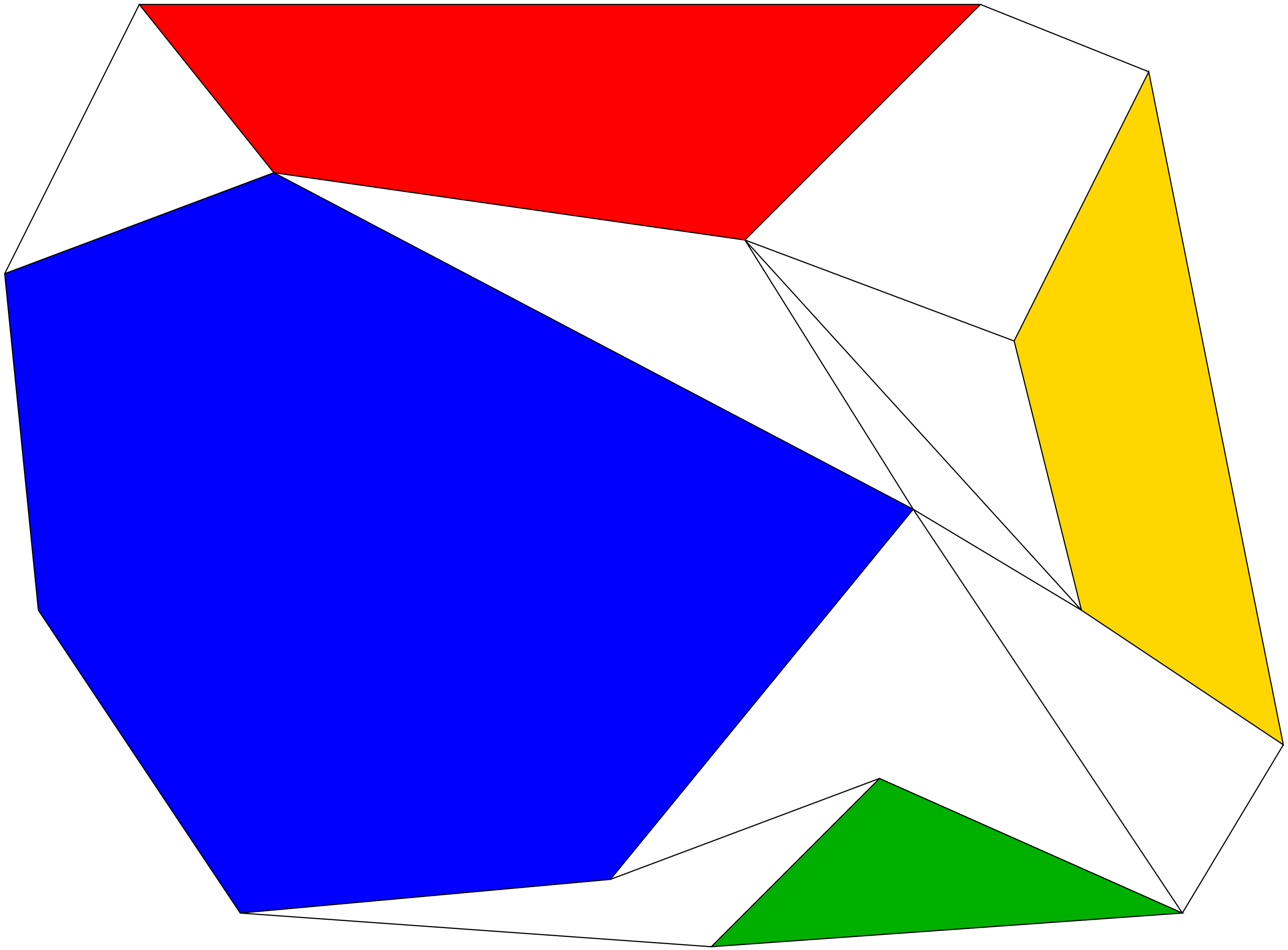
Ein irreduzibles Beispiel

Ein “echtes” planares Beispiel

FACETTENÜBERDECKUNG (FC)
Eingabe: ein eingebetteter planarer Graph $G = (V, E)$
Parameter: eine natürliche Zahl k
Frage: Gibt es eine Facettenüberdeckung der Größe höchstens k ?

Hierbei ist eine Facettenüberdeckung eine Menge von Facetten, an deren Grenzlinien alle Knoten des vorgelegten eingebetteten Graphen liegen.





Idee & Problem

Verzweigen nach “kleingradigen” Knoten scheint zunächst möglich nach der Euler-Formel.

ABER: wir erhalten schon nach dem ersten Verzweigen zwei Arten von Knoten: bereits abgedeckte (marked) und noch abzudeckende (active).
Facetten, auf denen wir bereits verzweigt haben, markieren wir ebenso.
Wir brauchen aber doch nicht auf kleingradigen markierten Knoten zu verzweigen ?!

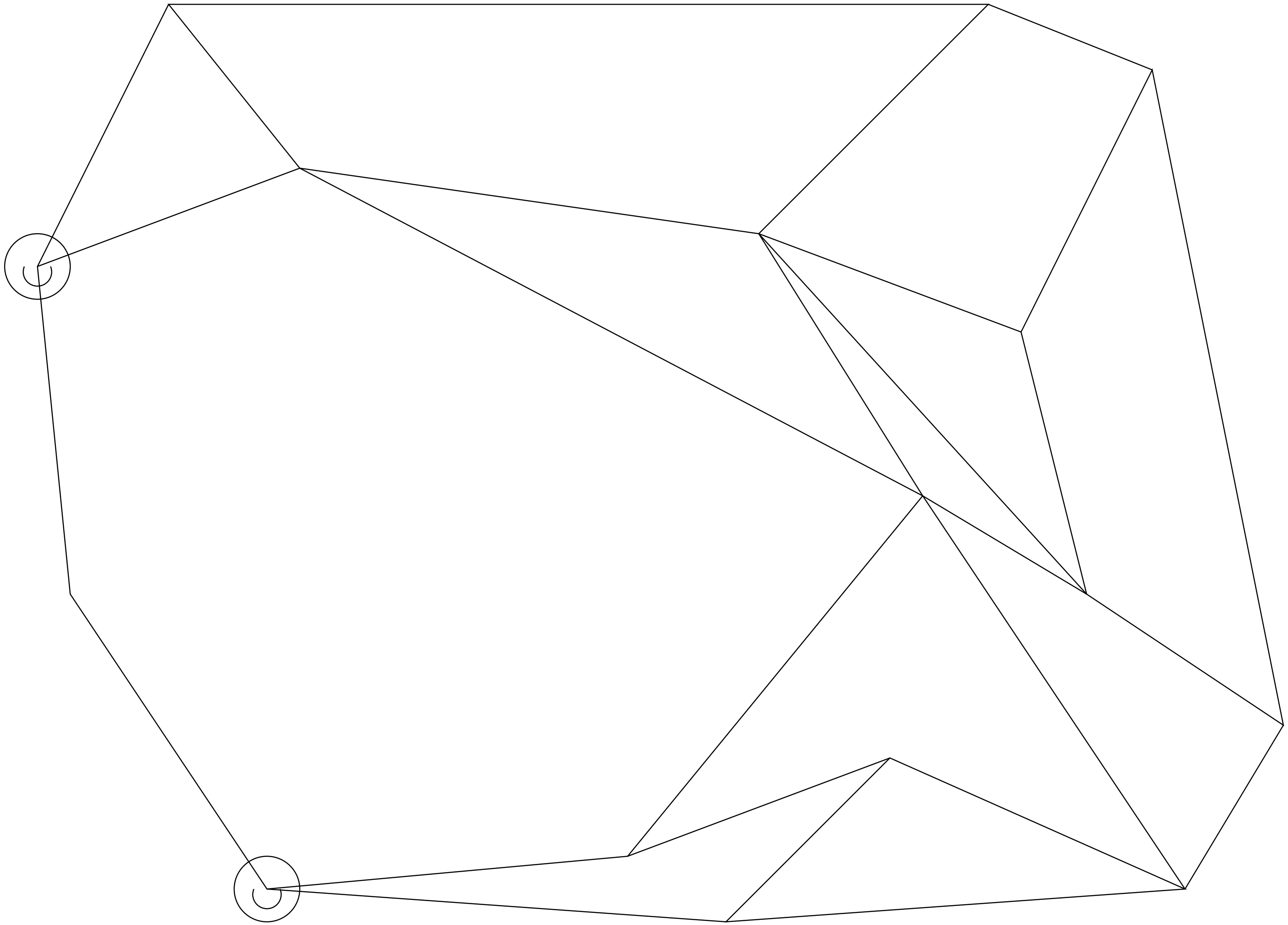
$\leadsto$ Annotierte Facettenüberdeckung, formalisiert durch Funktionen
$\mu_V : V \to \{active, marked\}$ und $\mu_F : F \to \{active, marked\}$.

Eingangs sind alle Knoten und Facetten aktiv.

HS-Regeln als Erbschaft

$F_a(v)$: die zu v benachbarten aktiven Facetten
$V_a(f)$: die an f anliegenden aktiven Knoten

— Gilt $F_a(u) \subseteq F_a(v)$ für aktive Knoten u, v, so markiere v.
—Gilt $|F_a(v)| = 1$ und ist v aktiv, so muss $F_a(v)$ in die Facettenüberdeckung; anschließend werden v und f markiert.
—Gilt $V_a(f) \subseteq V_a(f')$ für aktive Facette f, f', so markiere f.



Geometrische Operationen

—Sind u und v zwei benachbarte markierte Knoten, so verschmelze u und v (wobei möglicherweise Mehrfachkanten oder Schlingen eingeführt werden).

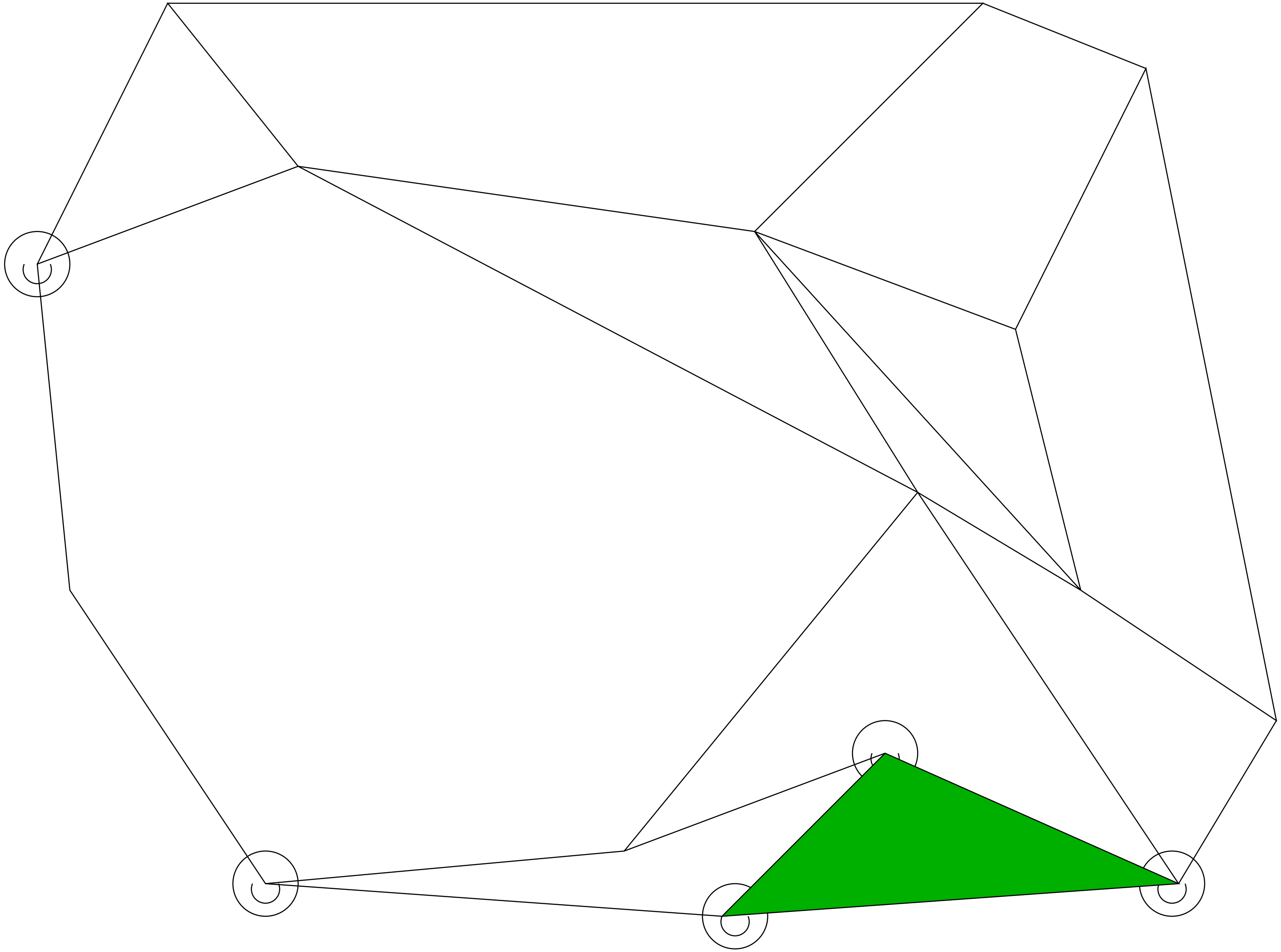
—Ist u ein markierter Knoten mit zwei aktiven Nachbarn v, w, sodass alle drei Knoten an einer aktiven Facette f anliegen, so unterteile f in zwei Facetten durch Einführung einer neuen Kante zwischen v und w, die innerhalb von f gezeichnet werden muss (wieder kann dies Mehrfachkanten liefern). Das neue Dreieck u, v, w ist markiert, während "der Rest" von f aktiv ist.

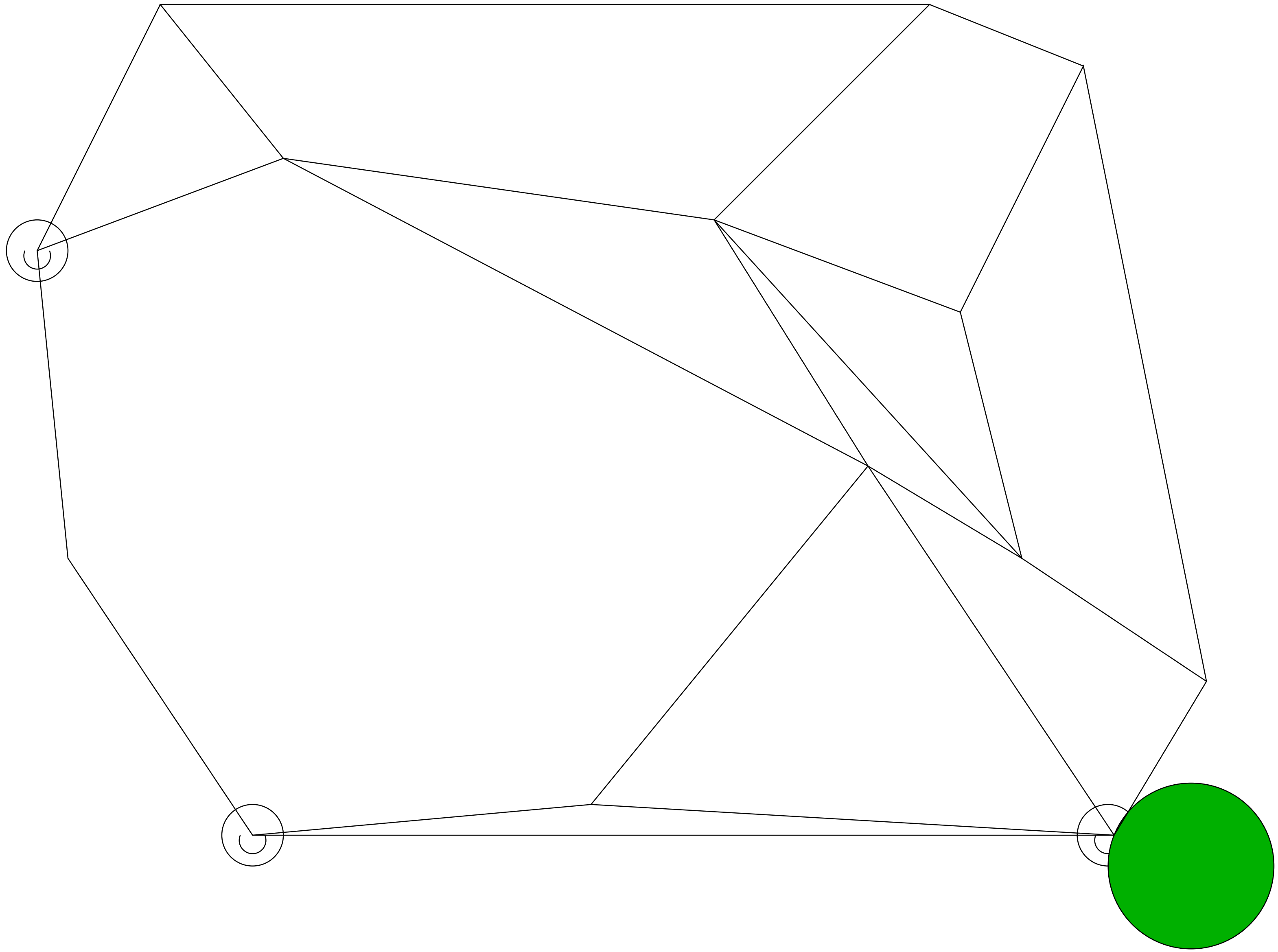
—Lösche markierte Knoten vom Grad Eins.

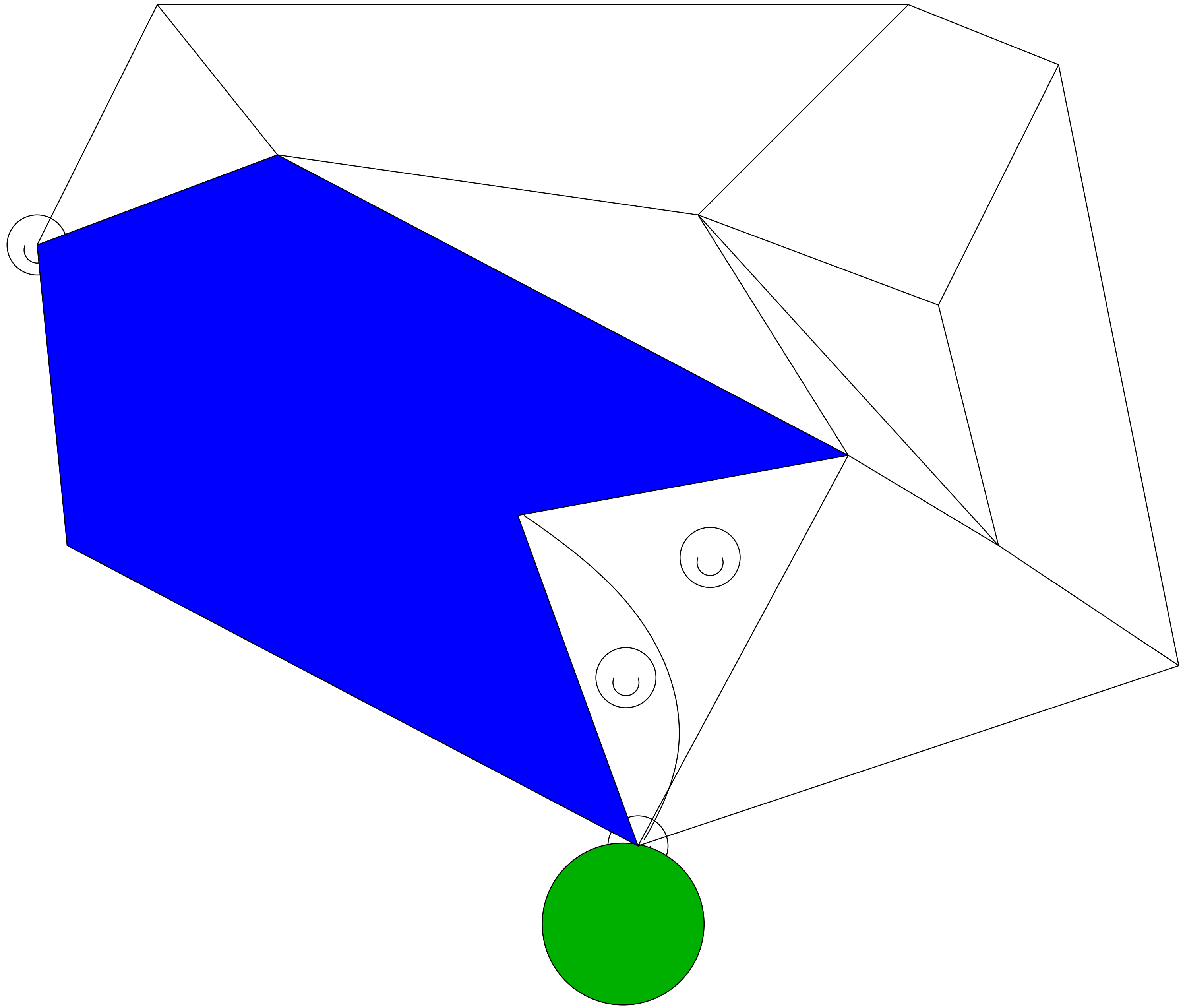
—Markierte Knoten v mit $|F_a(v)| = 0$ können ebenfalls gelöscht werden. So werden die v zuvor umschließenden markierten Facetten durch eine einzige markierte Facette ersetzt.

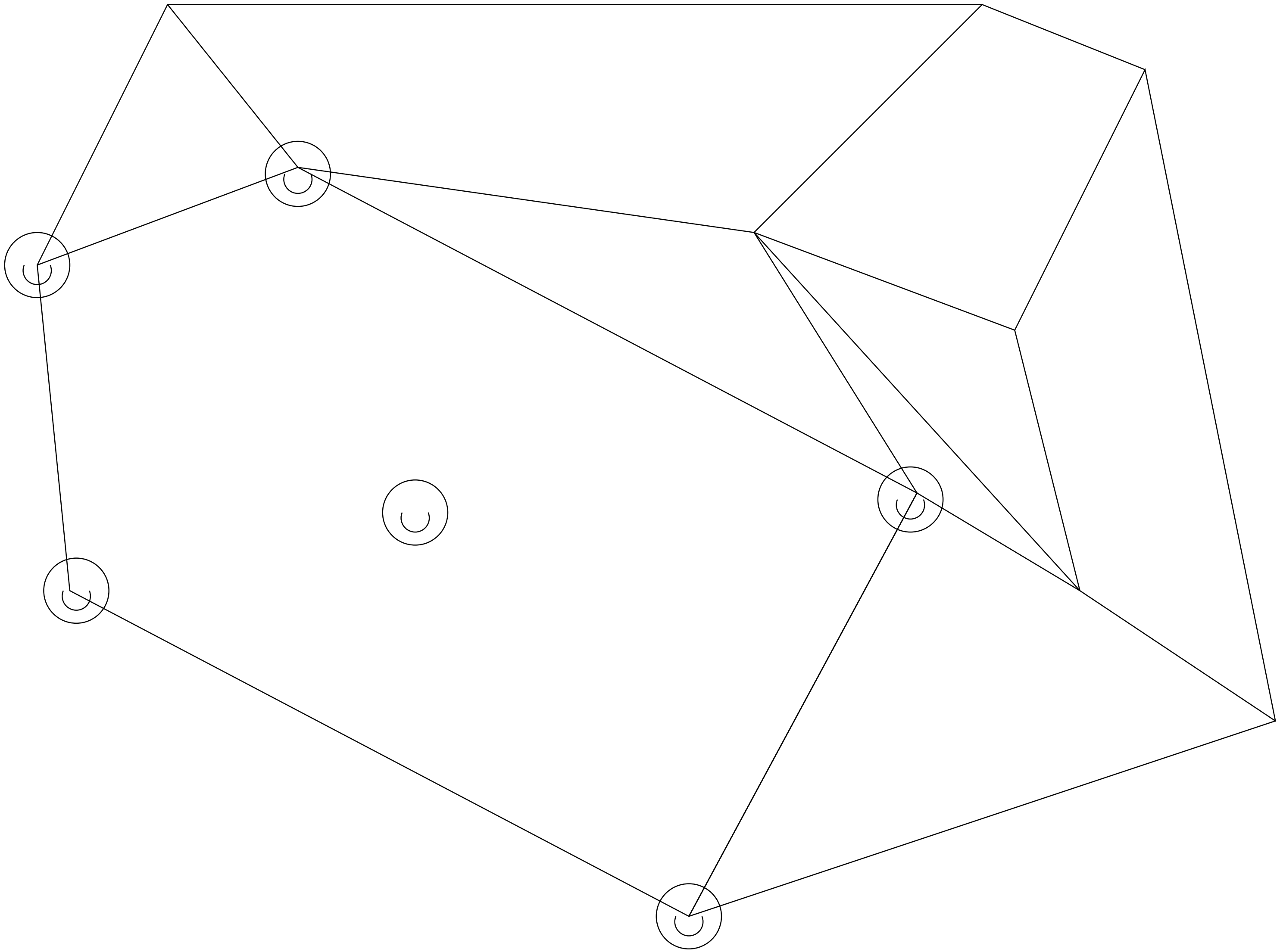
—Eine markierte Facette mit nur einem oder zwei Knoten auf ihrer Grenzlinie wird durch Löschen einer beliebig gewählten Begrenzungskante zerstört.

—Zwei markierte Facetten mit einer Grenzkante werden durch Löschen derselben zu einer neuen markierten Facette (u.U.) verschmolzen.









Algorithm 3 FC-ST: A simple search tree algorithm for annotated FC

Input(s): an annotated plane graph $G = (V, E)$ with face set F and marking functions μ_V and μ_F, a positive integer k

Output(s): YES iff G has an annotated face cover set $C \subseteq F$ with $|C| \leq k$

Exhaustively apply the reduction rules. {$\rightsquigarrow G$ (etc.) as before.}
if $k < 0$ **then**
  return NO
else if $V_a = \emptyset$ **then**
  return YES
else
  Let v be a vertex of lowest face degree in G.
  Choose $f \in F_a$ such that f is incident to v.
  Mark f and all vertices that are on the boundary of f.
  Call the resulting marking functions μ'_V and μ'_F.
  if FC-ST$(G, F, \mu'_V, \mu'_F, k - 1)$ **then**
    return YES
  else
    Mark f, i.e., return FC-ST(G, F, μ_V, μ'_F, k)

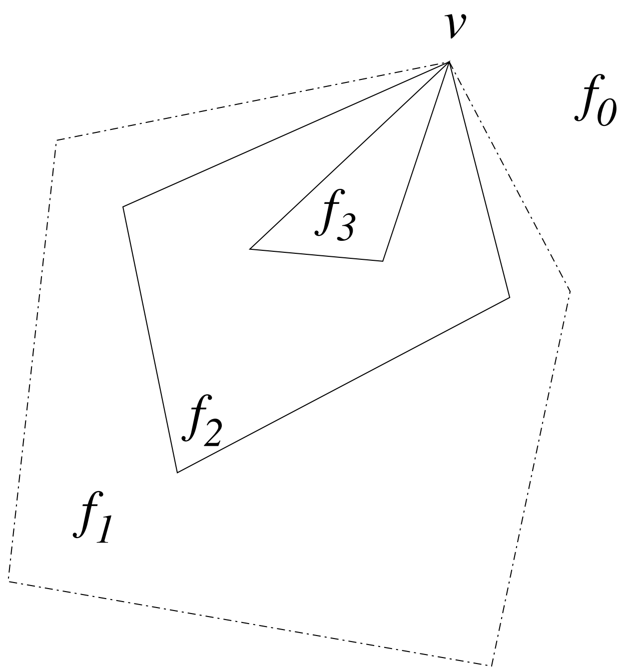
Satz 5 *Algorithmus 3 löst das AFC Problem in Zeit* $\mathcal{O}^*(5^k)$.

Zu zeigen wäre: In einem annotierten reduzierten (Multi-)graphen G gibt es stets einen Knoten, der an höchstens fünf aktiven Facetten anliegt.

Der Satz von Borodin u.a. gestattet wiederum, die Laufzeit noch deutlich zu verbessern.

Satz 6 *FC kann in Zeit* $\mathcal{O}^*(4.6056^k)$ *gelöst werden.*

Was sind Inseln ?



Im Beispiel: Am markierten Knoten v sind drei Schlingen (Multigraph!), die die Ebene in vier Bereiche (Inseln) aufteilen, die unabhängig voneinander gelöst werden können, da v markiert.

Weitere Aussagen

Lemma 7 *Ist $G = (V, E, V_a, F_a)$ ein annotierter, eingebetteter, reduzierter Multigraph ohne Inseln mit Facettenmenge F (als Instanz eines* ANNOTATED FACE COVER *Problems mit Parameter k), so gilt:*

1. *Es gibt keinen markierten Knoten in G.*

2. *Die einzigen degenerierten Facetten die möglicherweise existieren sind aktive Facetten f mit zwei inzidenten Knoten.*

3. *Jedes Knotenpaar $\{u, v\}$ hat höchstens eine inzidente degenerierte Facette.*

4. *Die beiden zu einer degenerierten Facette über Kanten angrenzenden Facetten sind markiert.*

Zum Beweis des “einfachen” Satzes:

Zu zeigen: Ist $G = (V, E, V_a, F_a)$ ein annotierter, eingebetteter, reduzierter Multigraph ohne Inseln mit Facettenmenge F (als Instanz eines ANNOTATED FACE COVER Problems mit Parameter k) gibt es stets einen Knoten mit höchstens fünf inzidenten aktiven Facetten.

Betrachte den schlichten eingebetteten Graphen $G' = (V, E')$, der aus G entsteht, indem wir (nur) eine Kante zwischen u und v zeichnen, sobald es irgendeine Kante zwischen u und v in G gibt.
Euler $\leadsto$ In G' können wir einen Knoten vom Maximalgrad fünf finden.
Gemäß dem Lemma könnten der Kante uv in G zwei Kanten (aber nicht mehr) zwischen u und v entsprechen. Somit gäbe es höchstens 10 an v angrenzende Facetten.
Von denen können aber (mit dem Lemma) höchstens 5 aktiv sein.

Algorithm 4 An advanced search tree algorithm for ANNOTATED FACE COVER, called FC-ST-advanced

Input(s): an annotated plane graph $G = (V, E)$ with face set F and marking functions μ_V and μ_F, a positive integer k

Output(s): C if G has an annotated face cover set $C \subseteq F$ with $|C| \leq k$; NO otherwise

```
Exhaustively apply the reduction rules.
{The resulting instance will be also called G (etc.) as before.}
if k < 0 then
  return NO
else if V_a = ∅ then
  return ∅
else if G contains a marked vertex v whose only neighbor is v itself then
  Recursively compute minimum face covers C_0, ..., C_r of the island graphs G_0, ..., G_r
  if ∃i : C_i = NO ∨ Σ_{i=0}^{r} |C_i| > k then
    return NO
  else
    return ⋃_{i=0}^{r} C_i
else
  {Details on next slide}
```

Algorithm 5 The non-island-case

Let u be a vertex of lowest face degree in G.
{One incident face of u must be used to cover u.}
if $\deg_a(u) \leq 4$ **then**
  Choose $f \in F_a$ such that f is incident to u.
  Mark f and all vertices that are on the boundary of f.
  Call the resulting marking functions μ'_V and μ'_F.
  Let C :=FC-ST-advanced($G, F, \mu'_V, \mu'_F, k-1$)
  if $C \neq$ NO **then**
    return $C \cup \{f\}$
  else
    Mark (only) f, i.e., return FC-ST-advanced(G, F, μ_V, μ'_F, k)
else
  {Let $N(u) = \{u_1, u_2, v, w, z\}$ be the neighbors of u and similarly $N(v) = \{v_1, v_2, v_3, u, w, z\}$.}
  if all active faces incident with v are degenerate **then**
    execute FC-ST-case-1
  else if no active faces incident with v are degenerate **then**
    execute FC-ST-case-2
  else
    {all active faces incident with u are degenerate; only one active face incident with v is degenerate}
    execute FC-ST-case-3

Algorithm 6 The code of FC-ST-case-1

Let $G' = G \setminus \{u, v\}$ and mark the face f to which (formally) u, v belonged;
modify F, μ_V, μ_F accordingly, yielding F', μ'_V and μ'_F.
Let C :=FC-ST-advanced($G', F', \mu'_V, \mu'_F, k-1$)
if $C \neq$ NO **then**
  return $C \cup \{f\}$
else
  for all unordered vertex pairs $\{x, y\}$ such that $x \in N(u) \setminus \{v\}$ and $y \in N(v) \setminus \{x, u\}$ **do**
    Modify μ'_V so that x and y are the only vertices of $N(u) \cup N(v) \setminus \{u, v\}$ that are marked.
    Let C :=FC-ST-advanced($G', F', \mu'_V, \mu'_F, k-2$)
    if $C \neq$ NO **then**
      return $C \cup \{f\}$
  return NO

Algorithm 7 The code of FC-ST-case-2

```
Let G′ = G \ {u, v} and mark the face f to which (formally) u, v belonged;
modify F, μ_V, μ_F accordingly, yielding F′, μ′_V and μ′_F.
for all vertices x ∈ {w, z} do
    Modify μ′_V so that x is the only vertex of N(u) ∪ N(v) \ {u, v} that is marked.
    Let C :=FC-ST-advanced(G′, F′, μ′_V, μ′_F, k − 1)
    if C ≠ NO then
        return C ∪ {f}
for all vertices x ∈ {u_1, u_2} and y ∈ {v_1, v_2, v_3} do
    Modify μ′_V so that x and y are the only vertices of N(u) ∪ N(v) \ {u, v} that are marked.
    Let C :=FC-ST-advanced(G′, F′, μ′_V, μ′_F, k − 2)
    if C ≠ NO then
        return C ∪ {f}
return NO
```

Algorithm 8 The code of FC-ST-case-3

```
for all faces f ∈ F_a(u), g ∈ F_a(v) do
  Mark f and g and all vertices in V_a(f) ∪ V_a(g); call the modified marking functions μ'_V and μ'_F.
  Let C :=FC-ST-advanced(G', F, μ'_V, μ'_F, k − |{f, g}|)
  if C ≠ NO then
    return C ∪ {f, g}
return NO
```

Hinweis: Face Cover wird gerade im Rahmen einer Diplomarbeit implementiert.

Zusammenfassung Suchbaumentwurf

Grundidee: Gibt es beschränkte Kandidatenmengen, aus denen parameterreduzierend ein Element ausgewählt werden muss ?

Beispiele:
Hyperkante bei Hitting Set
Knoten-Nachbarschaft bei PIS

Dabei (wie so) oft hilfreich: mathematische Ergebnisse
Bsp.: Existenz kleingradiger Knoten in planaren Graphen

Noch ein Beispiel: Wie ergibt sich hier ein Suchbaumalgorithmus ?
Eine Clique ist ein vollständiger Teilgraph.
Ein Cluster ist eine Clique, die eine Zusammenhangskomponente ist.
Bei einem Clustergraph bildet jede Zusammenhangskomponente eine Clique.

CLUSTERKNOTENPROBLEM (CLUSTER VERTEX DELETION CVD)
Eingabe: Ein Graph $G = (V, E)$
Parameter: $k \in \mathbb{N}$
Frage: Gibt es eine Knotenmenge $D \subseteq V$, sodass $|D| \leq k$ und $G - D$ ist ein Clustergraph ?

Lemma 8 *Enthält G einen induzierten P_2, so ist G kein Clustergraph (und umgekehrt).*

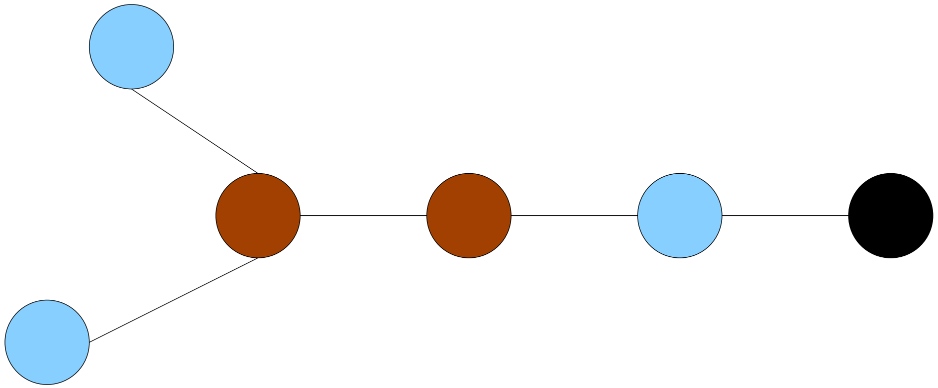
Frage: Wie sehen also geeignete Kandidatenmengen aus (zum Verzweigen) ?

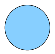
$\boxed{\textbf{Zusammenfassung}}$ der Grundtechniken

Kerne (Datenreduktion) kennzeichnen FPT
Praktisch sind sie alleine eher Heuristiken zum Lösen harter Probleme.
In Kombination mit guten Suchbaumstrategien liefert dies die praktisch besten parameterisierten Algorithmen.

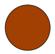
Suchbäume alleine scheinen nicht FPT zu kennzeichnen (offene Frage !).
Suchbäume liefern meist den enscheidend(verbessernd)den Anteil für gute Laufzeitabschätzungen.

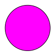
Parameterisierung nach $n = |V|$:
Berechnung kleinstmöglicher dominierender Mengen



: in der dominierenden Menge

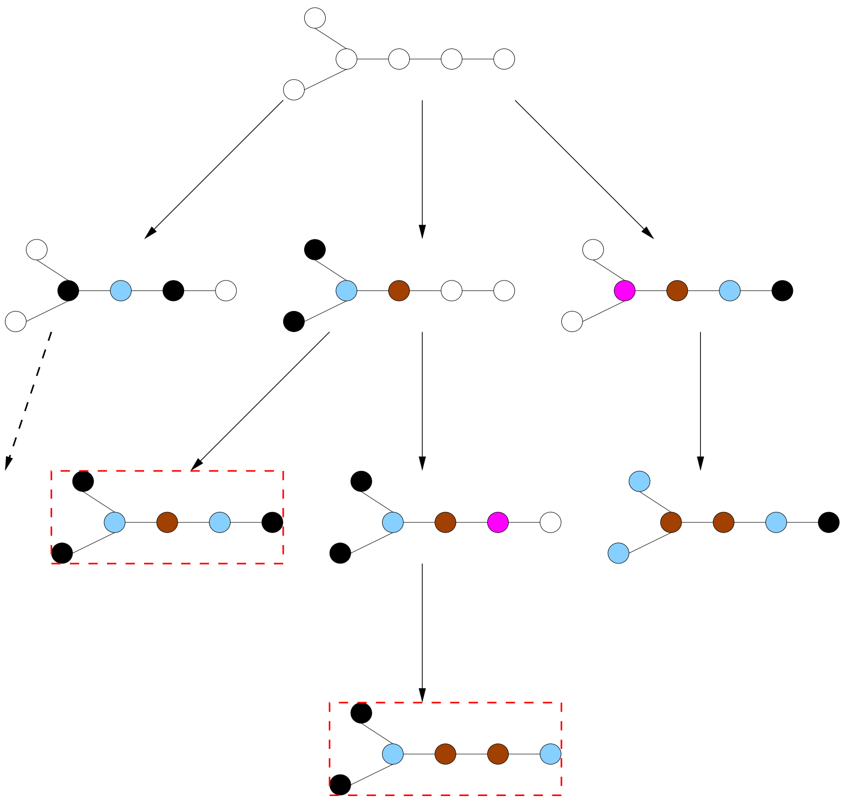
: dominiert

: dominiert; soll nicht in die dominierende Menge

: noch nicht dominiert; soll nicht in die dominierende Menge

Ist diese Lösung kleinstmöglich?

Ein farbenfroher Suchbaum



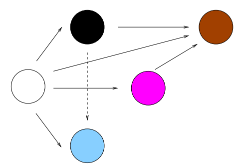
: noch unentschieden

: in der dominierenden Menge

: dominiert

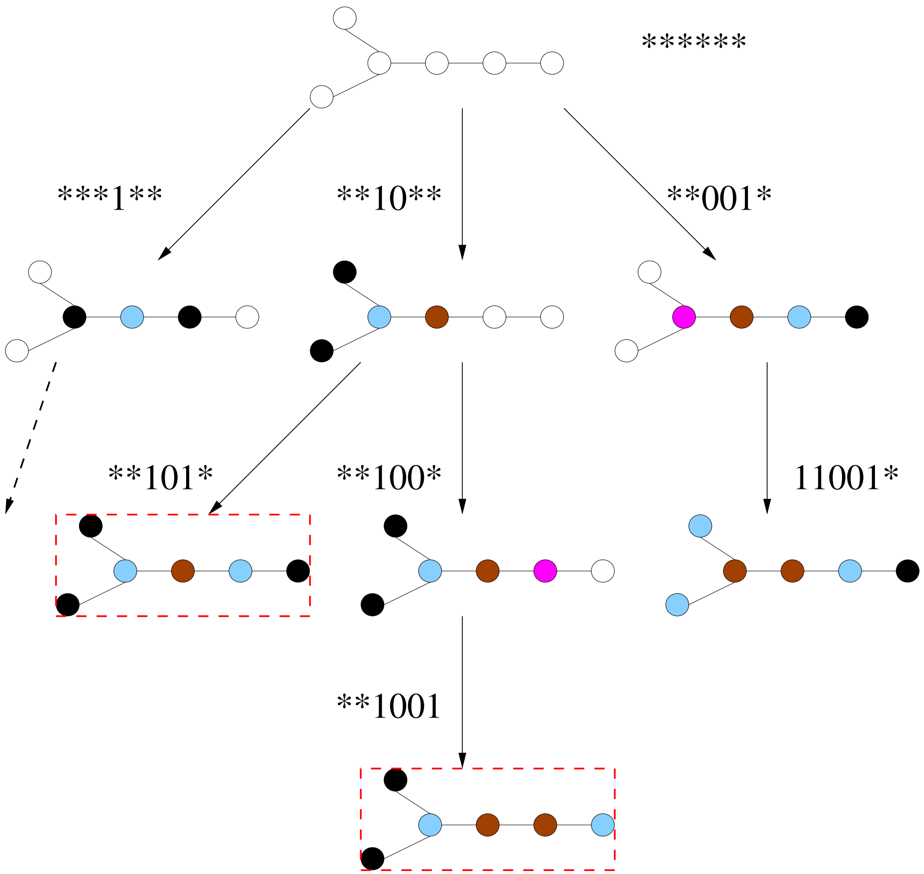
: dominiert; soll nicht in die dominierende Menge

: noch nicht dominiert; soll nicht in die dominierende Menge

Mögliche Übergänge: 

Verzweige gemäß oder Knoten, solange es oder gibt.

Ein farbenfroher Suchbaum



Suchbaum
mit Knotenbeschriftungen.
Beschriftungen kodieren
Verzweigungsentscheidungen.
Je näher zu den Blättern, desto genauer sind die Beschriftungen.
Beobachte: Selbst an den Blättern bestehen die Beschriftungen nicht ausschließlich aus 0 oder 1.

Trivialer Suchbaum?!

Naheliegendes Verzweigen auf Knoten:
0: nicht in dominierender Menge
1: kommmt in dominierende Menge
Das liefert triviale Abschätzung 2^n der Suchbaumgröße.

Beobachte: Niemals nutzen wir aus, dass wir auch Nachbarn “erobern”, so wir Knoten in die dominierende Menge einfügen.

Offenbar von Vorteil:
Verzweigen auf ○ im Ggs. zu ●.
Was ist, wenn es keine ○, aber noch ● und ● im Graph gibt?

Gibt es einen Punkt, an dem die Restinstanz in Polynomzeit gelöst werden kann?

Measure & Conquer für Dominating Set

Algorithm 9 A simple algorithm for Dominating Set

if possible choose a $v \in \bigcirc$ such that $|N(v) \cap (\bigcirc \cup {\color{magenta}\bullet})| \geq 2$; **then**
    Binary branch on v (i.e., set v ${\color{cyan}\bullet}$ in one branch, ${\color{magenta}\bullet}$ in the other)
else if possible choose a $v \in \bullet$ such that $|N(v) \cap (\bigcirc \cup {\color{magenta}\bullet})| \geq 3$; **then**
    Binary branch on v.
else
    Solve the remaining instance in polynomial time using an Edge Cover algorithm.

Measure & Conquer: DOMINATING SET mit Hilfe von EDGE COVER.

In Schritt 6 erzeuge EDGE COVER Instanz $G_{EC} = (V(E_{EC}), E_{EC})$:

(1a) Für $v \in \bigcirc$ mit $N(v) \cap (\bigcirc \cup \textcolor{magenta}{\bullet}) = \{q\}$,

füge $e = \{v, q\}$ zu E_{EC} und setze $\alpha(e) = v$,

(1b) für $v \in \bullet$ mit $N(v) \cap (\bigcirc \cup \textcolor{magenta}{\bullet}) = \{x, y\}$, $x \neq y$,

füge $e = \{x, y\}$ zu E_{EC} und setze $\alpha(e) = v$.

Ist C eine kleinstmögliche Kantenüberdeckung von G_{EC}, füge v zu DS, wobei $v = \alpha(e)$ für ein $e \in C$.

(2) Füge alle $v \in \bigcirc \setminus V(E_{EC})$ mit $N(v) \cap (\bigcirc \cup \textcolor{magenta}{\bullet}) = \emptyset$ in DS.

(3) Füge kein $v \in \bullet$ mit $|N(v) \cap (\bigcirc \cup \textcolor{magenta}{\bullet})| = 0$ in DS.

(4) Füge alle $v \in \bullet$ mit $N(v) \cap (\bigcirc \cup \textcolor{magenta}{\bullet}) = \{s\} \not\subseteq V(E_{EC})$ in DS.

DOMINATING SET: **Analyse der Verzweigungen mit dem Maß**

$\mu = |\bigcirc| + \omega \cdot (|\textcolor{magenta}{\bullet}| + |\bullet|) \leq n$

Sei $n_{\bigcirc} = |N(v) \cap \bigcirc|$ und $n_{\textcolor{magenta}{\bullet}} = |N(v) \cap \textcolor{magenta}{\bullet}|$.

Wenn v ins DS in **Schritt** 2 kommt, reduziere μ zunächst um 1 (v verschwindet).

$\rightsquigarrow$ Knoten in $N(v) \cap \bigcirc$ werden dominiert und kommen daher in die Menge $\bullet$.

$\rightsquigarrow$ μ wird um $n_{\bigcirc} \cdot (1 - \omega)$ reduziert.

Ähnlich: Knoten aus $N(v) \cap \textcolor{magenta}{\bullet}$ werden dominiert und verschwinden so aus μ.

$\rightsquigarrow$ μ wird um $n_{\textcolor{magenta}{\bullet}}\omega$ reduziert.

Kommt v nicht in DS, reduziere μ um $(1 - \omega)$: v wandert aus $\bigcirc$ in $\textcolor{magenta}{\bullet}$.

$\rightsquigarrow$ Verzweigungsvektor:

$$(1 + n_{\bigcirc}(1 - \omega) + n_{\textcolor{magenta}{\bullet}}\omega, (1 - \omega)) \tag{1}$$

mit $n_{\bigcirc} + n_{\textcolor{magenta}{\bullet}} \geq 2$ in Schritt 1.

Ähnliche Analyse für **Schritt** 4 liefert wegen $n_{\bigcirc} + n_{\textcolor{magenta}{\bullet}} \geq 3$ in Schritt 3:

$$(\omega + n_{\bigcirc}(1 - \omega) + n_{\textcolor{magenta}{\bullet}}\omega, \omega) \tag{2}$$

DOMINATING SET: **Analyse der Verzweigungen in den Schritten 2 und 4.**

Abhängig von $n_{\circ}$ und $n_{\bullet}$ erhalten wir eine *unendliche Zahl von Verzweigungsfällen*. $\rightsquigarrow$ Beschränkung auf die schlimmsten Fälle.

Für (1) sinddiese gegeben durch $n_{\circ} + n_{\bullet} = 2$ und für (2) $n_{\circ} + n_{\bullet} = 3$.

$\rightsquigarrow$ endliche Zahl von Rekurrenzen $R_1(\omega), \ldots, R_7(\omega)$ abhängig von ω.

Wähle ω so, dass die größte Nullstelle der zugehörigen charakteristischen Polynome minimal wird. In unserem Fall: $\omega := 0.5$.

Der schlimmste Verzweigungsvektor ist nun $(2, 0.5)$.

$\rightsquigarrow$ Die Blattanzahl im Suchbaum ist beschränkt durch $\mathcal{O}^*(1.9052^{\mu})$.

$\rightsquigarrow$ Der Algorithmus bricht die 2^n-Schranke mit einem einfachen Maß.