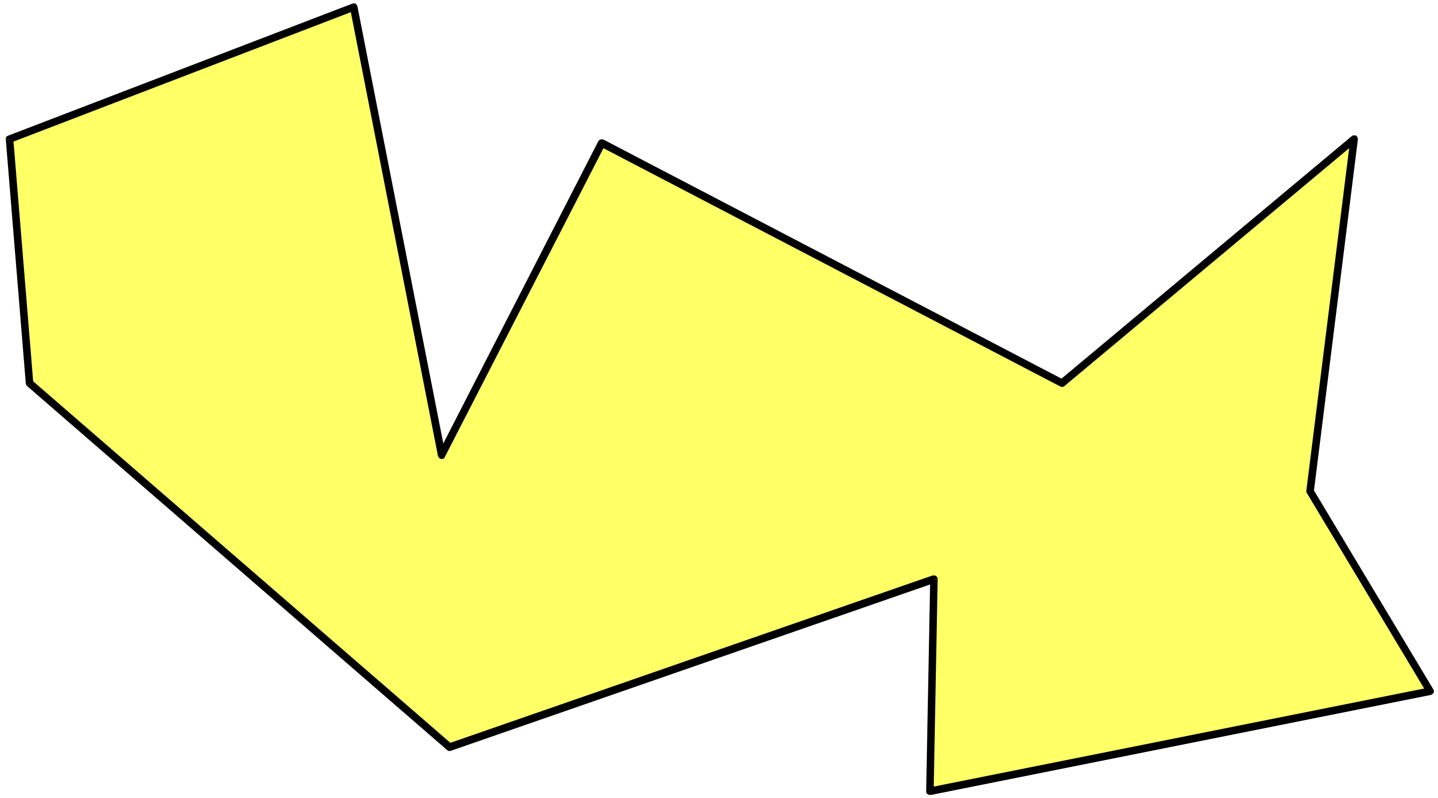
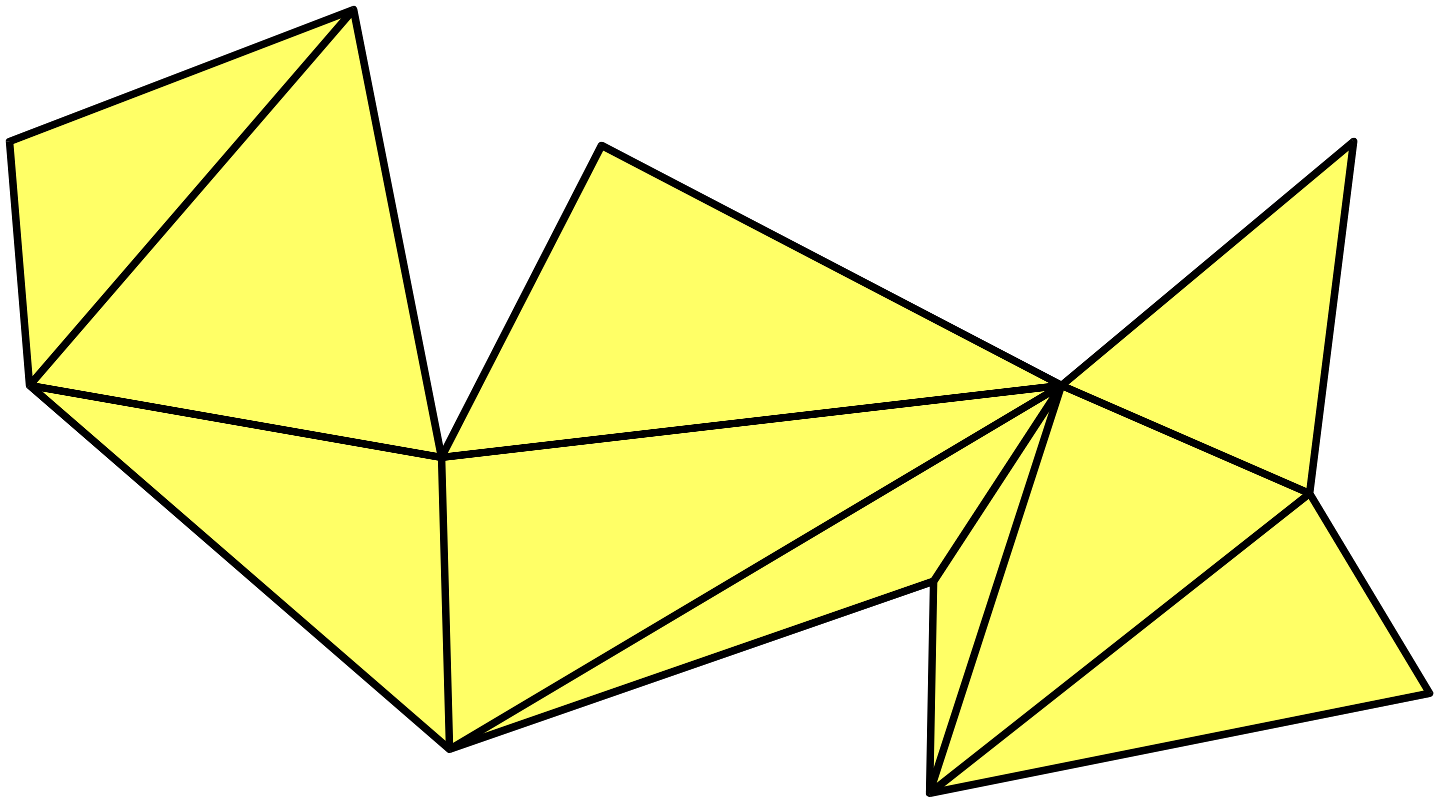
6. Triangulation von Polygonen

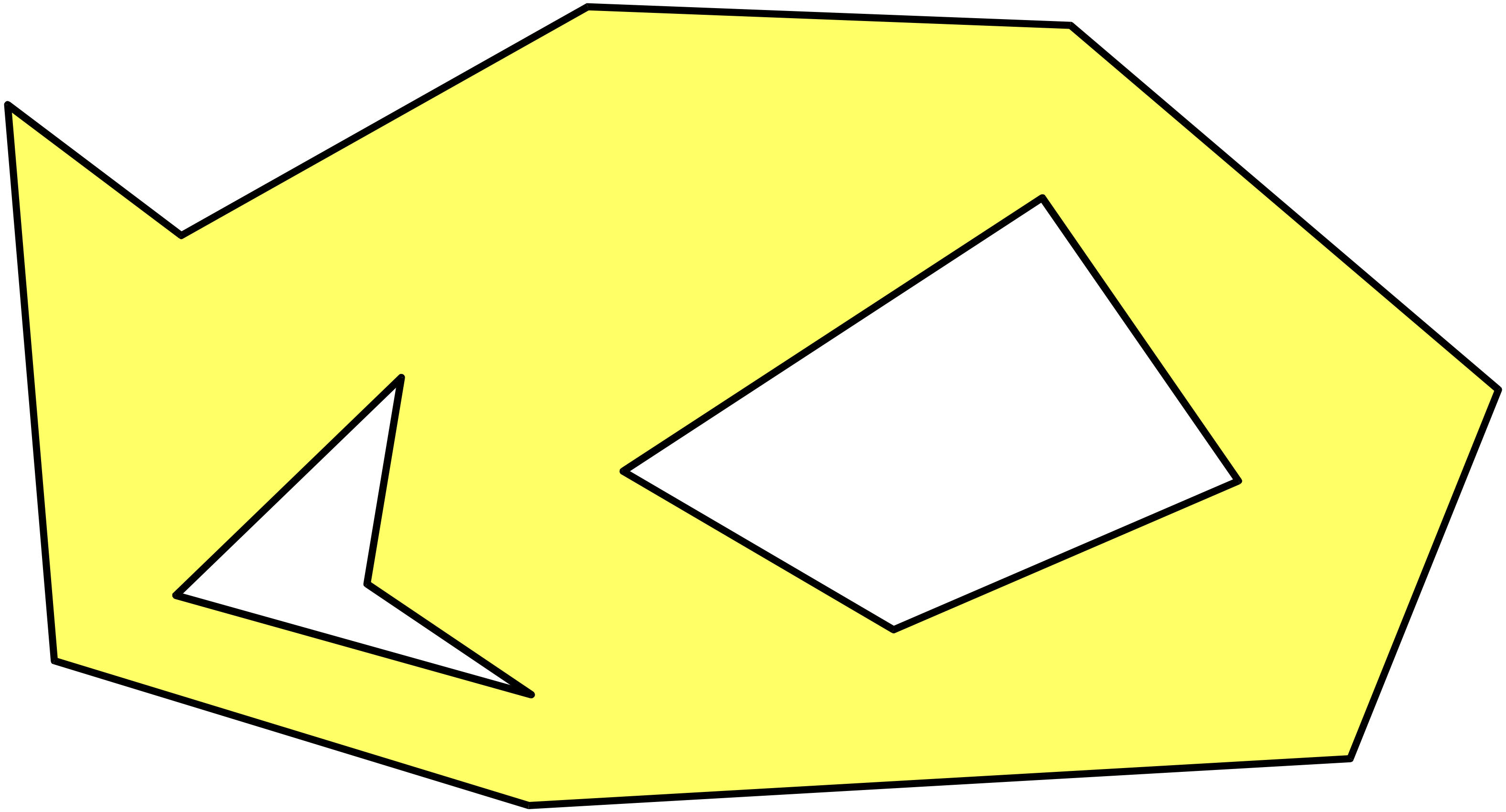
Problemstellung



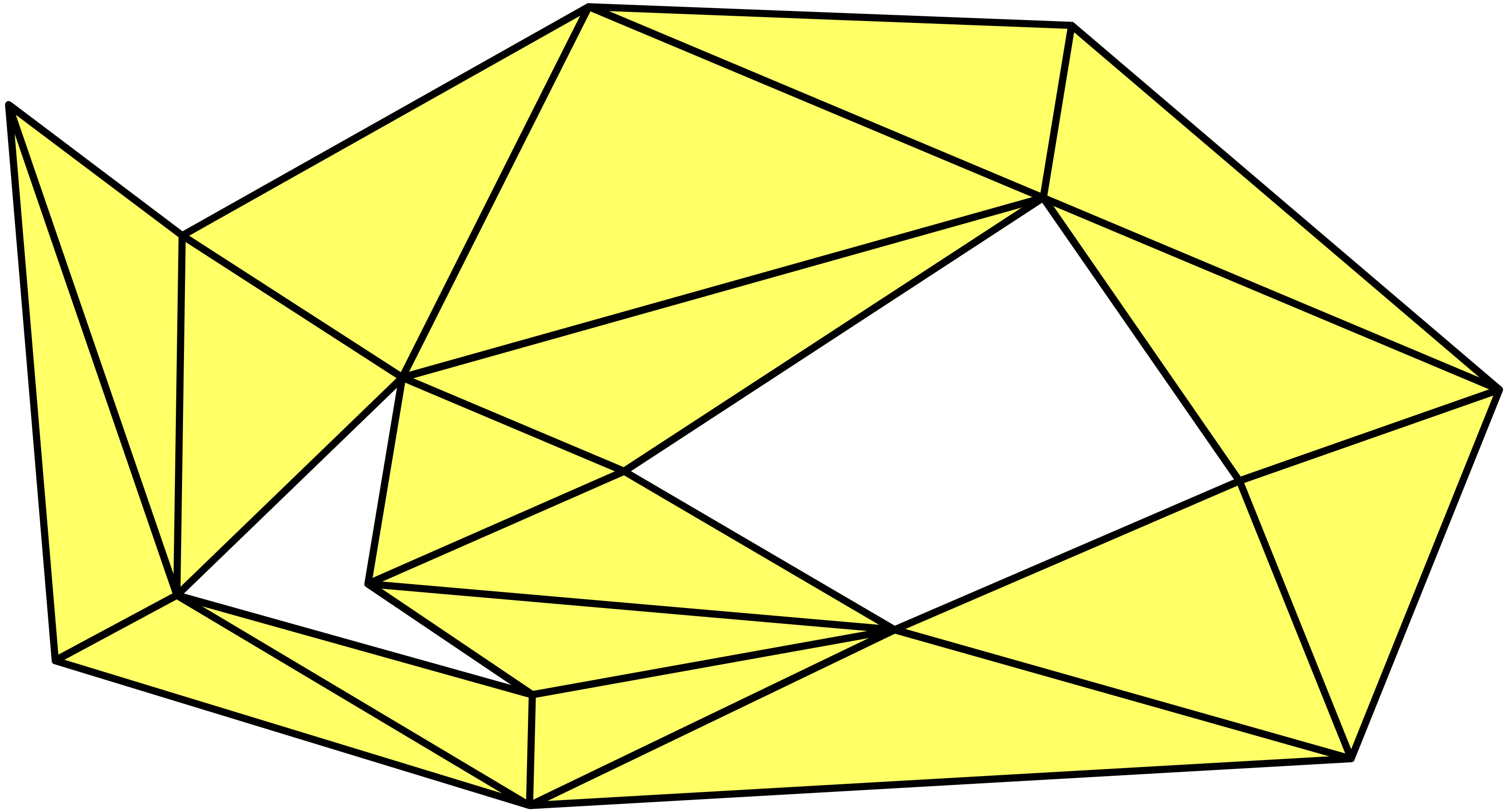
Problemstellung



Problemstellung



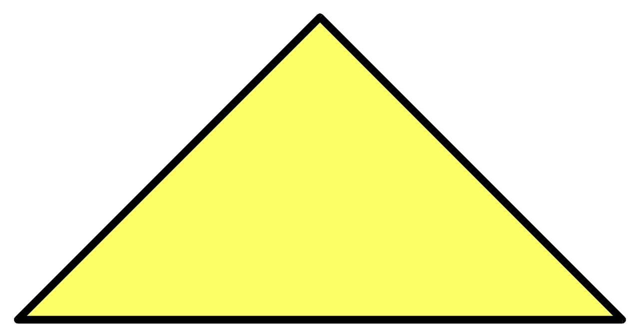
Problemstellung



Jedes Polygon lässt sich triangulieren.

Wir führen einen Induktionsbeweis nach der Anzahl der Eckpunkte n und Anzahl der Löcher h.

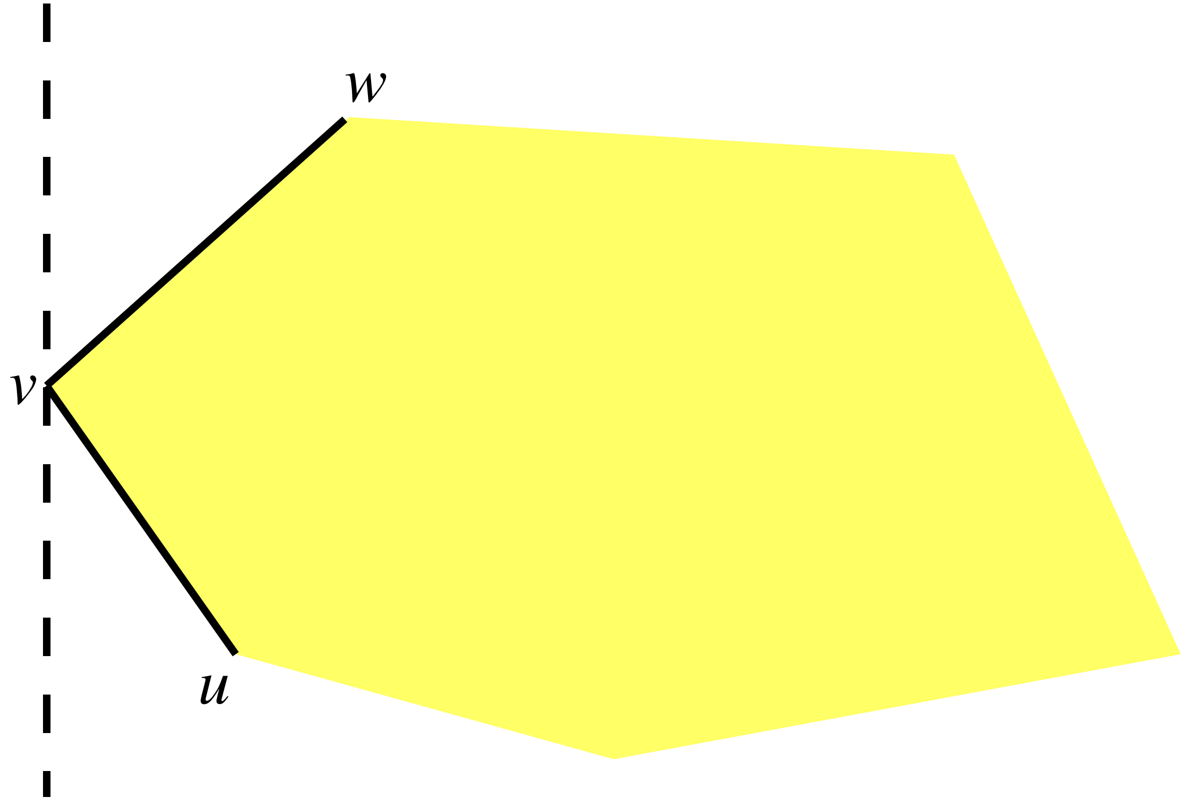
Induktionsanfang: $n = 3$ und $h = 0$



Jedes Polygon lässt sich triangulieren.

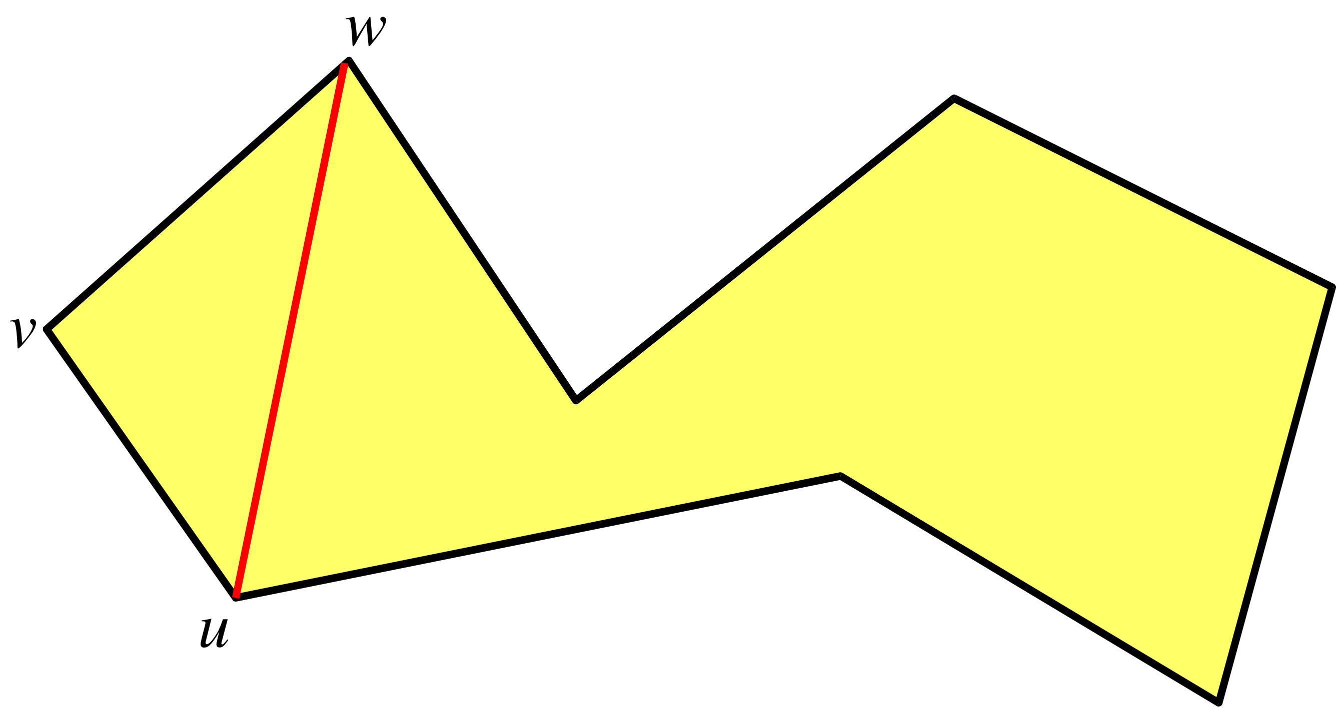
Induktionsschritt:

Wir bestimmen die Ecke v mit der kleinsten x-Koordinate und deren Vorgänger u und Nachfolger w.



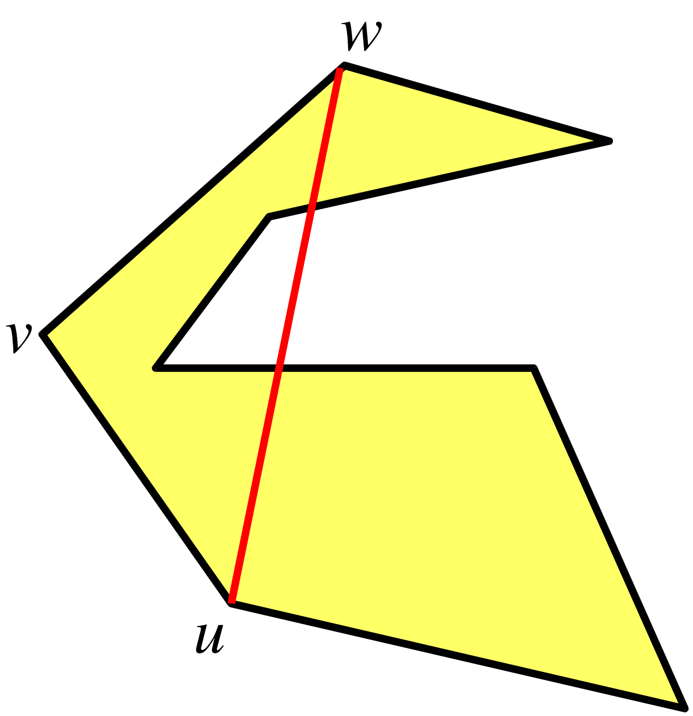
Jedes Polygon lässt sich triangulieren.

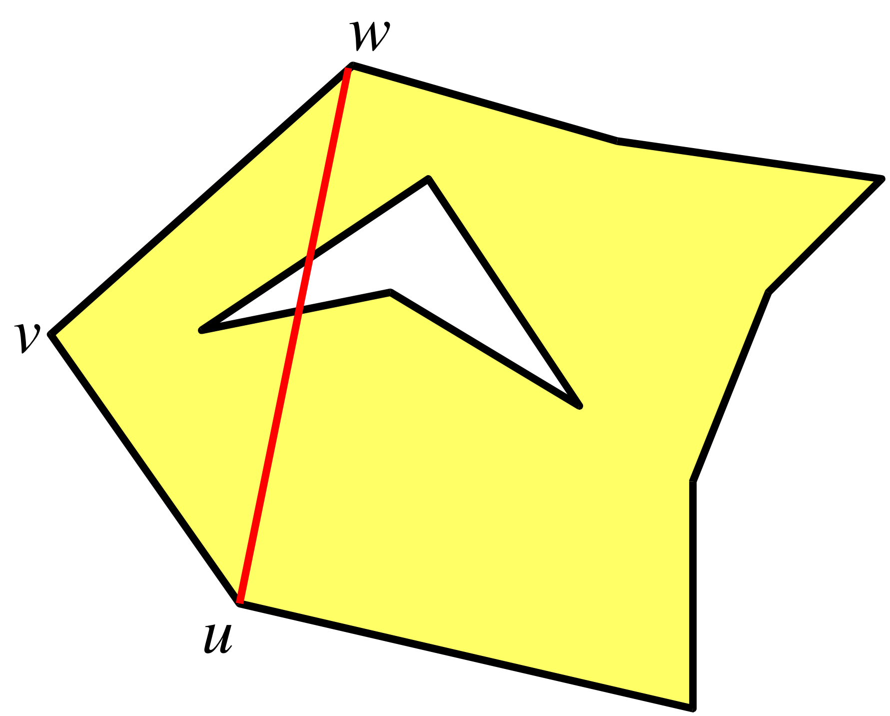
1. Fall: Die Strecke zwischen u und w verläuft innerhalb des Polygons.



Jedes Polygon lässt sich triangulieren.

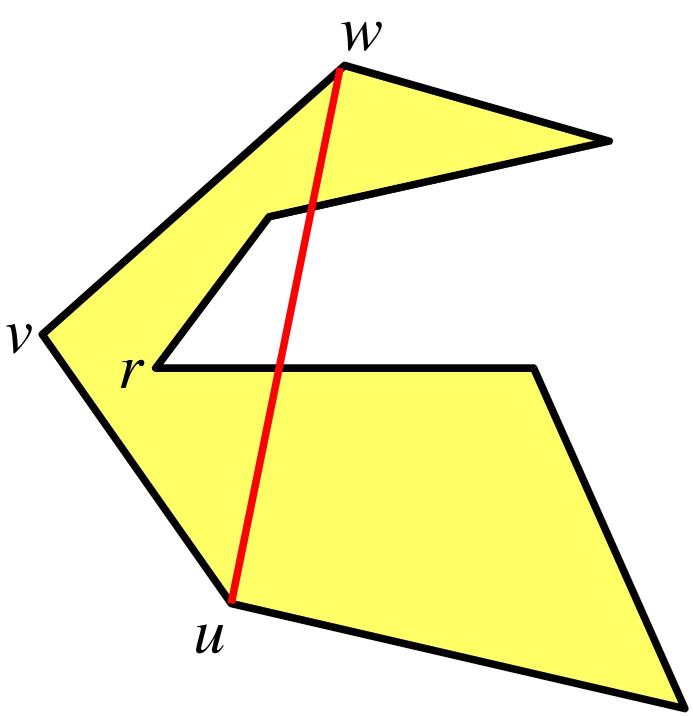
2. Fall: Die Strecke zwischen u und w verläuft nicht innerhalb des Polygons.

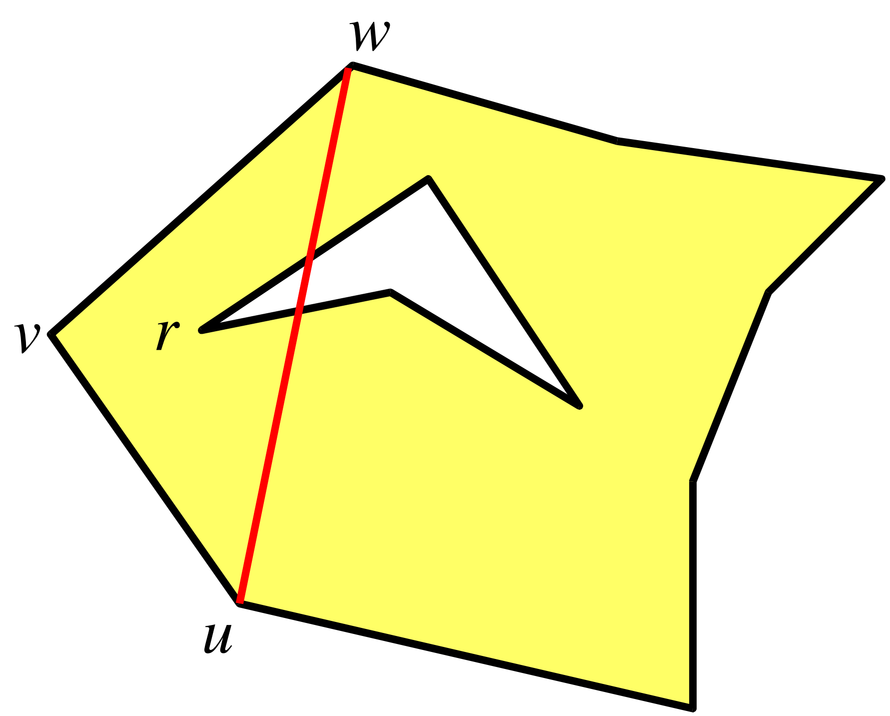




Jedes Polygon lässt sich triangulieren.

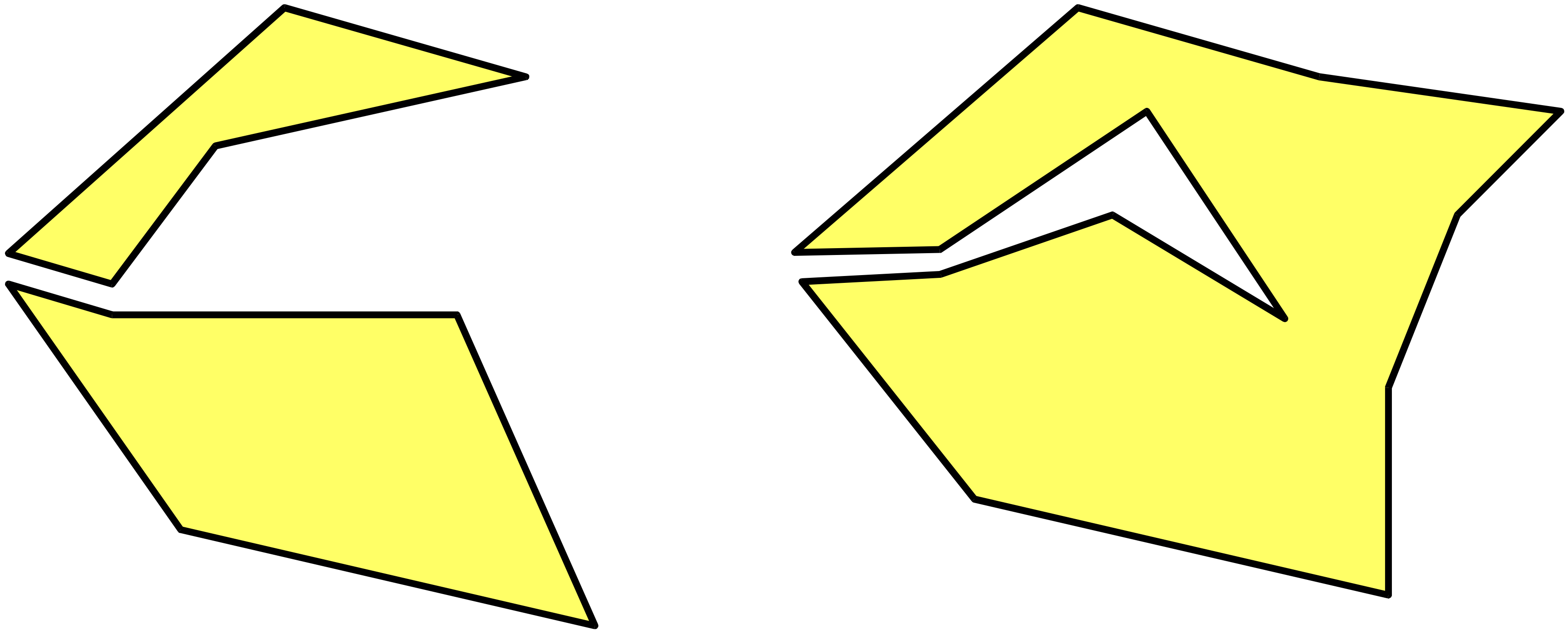
Wir bestimmen die Ecke r im Inneren des Dreiecks uvw mit kleinster x-Koordinate.





Jedes Polygon lässt sich triangulieren.

Wir zerlegen das Polygon entlang der Diagonale *vr*.



Bemerkungen

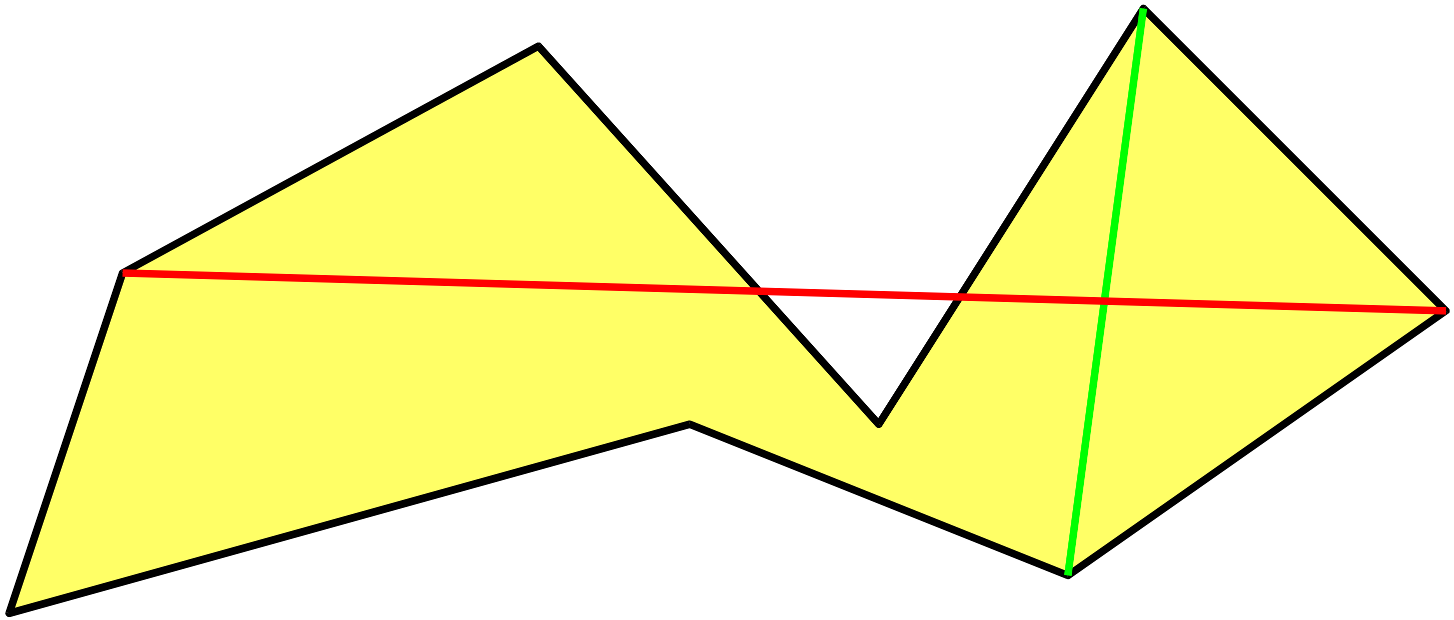
Der Induktionsbeweis führt unmittelbar zu einem Algorithmus mit Laufzeit $O(n^2)$.

Ferner gilt: SORTIEREN $\leq_{O(n)}$ TRIANGULIEREN

Der Beweis benutzt allerdings Polygone mit Löchern.

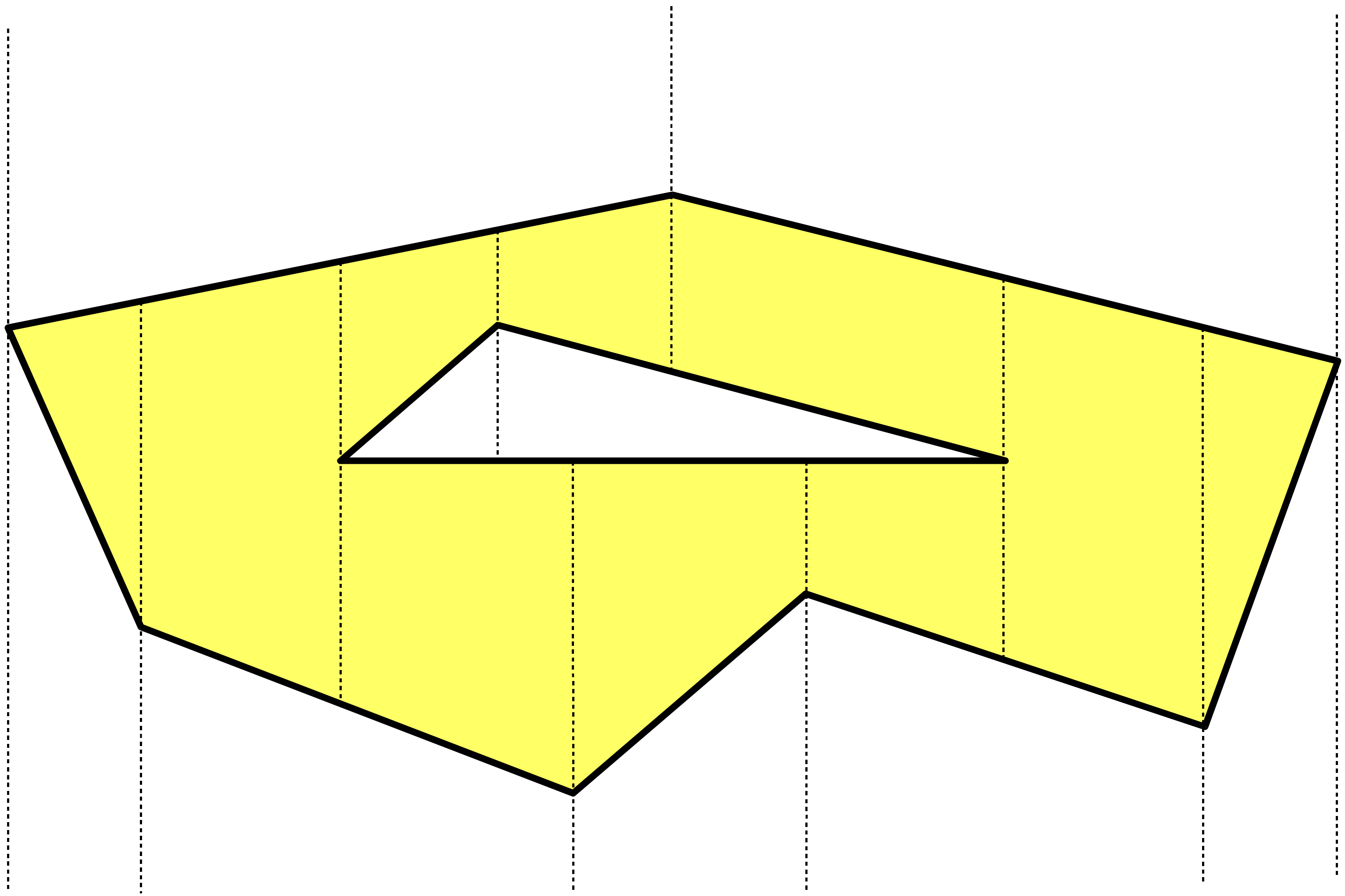
Beobachtung

Beim Triangulieren geht es offenbar darum, möglichst effizient Diagonalen im Polygon zu finden.



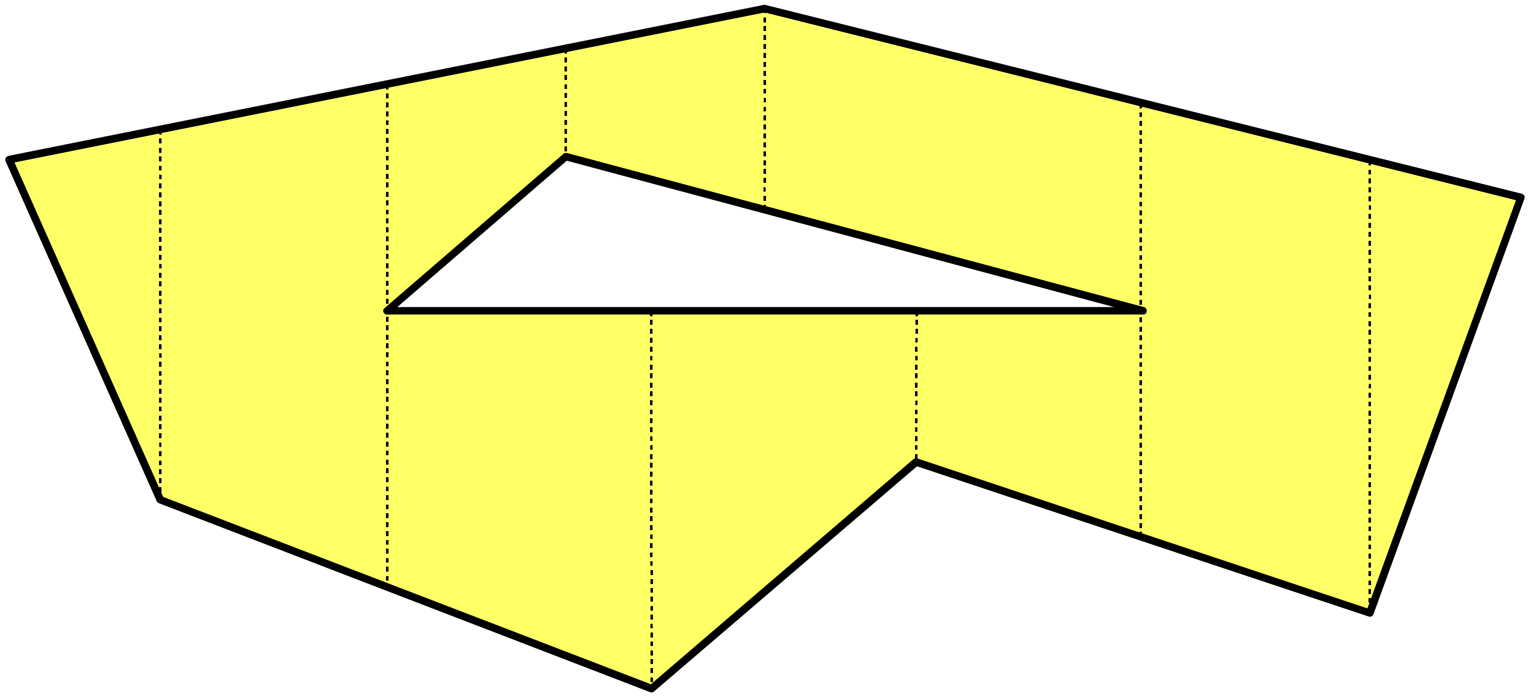
Benutzung der Trapezzerlegung

Wir betrachten die Trapezzerlegung für die Kanten des Polygons.



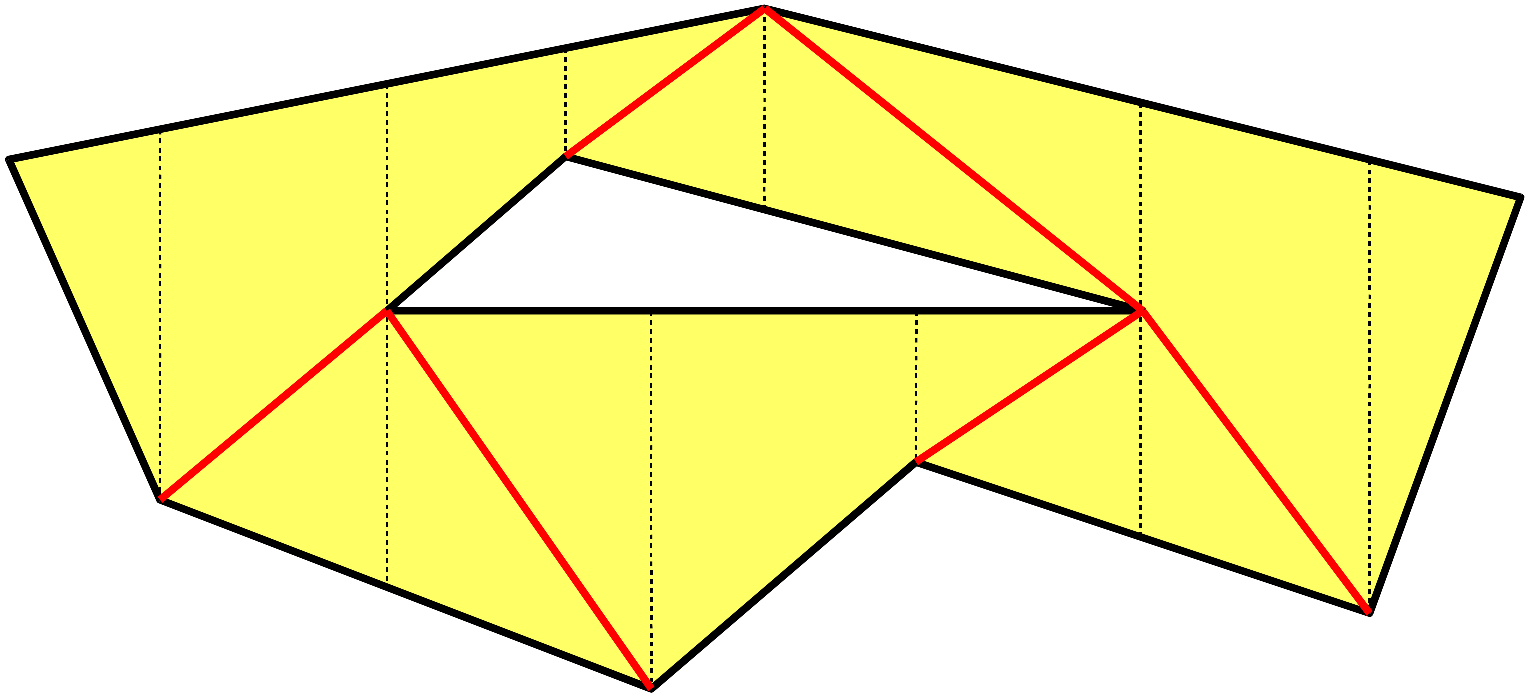
Benutzung der Trapezzerlegung

Alle Trapeze außerhalb des Polygons werden entfernt.



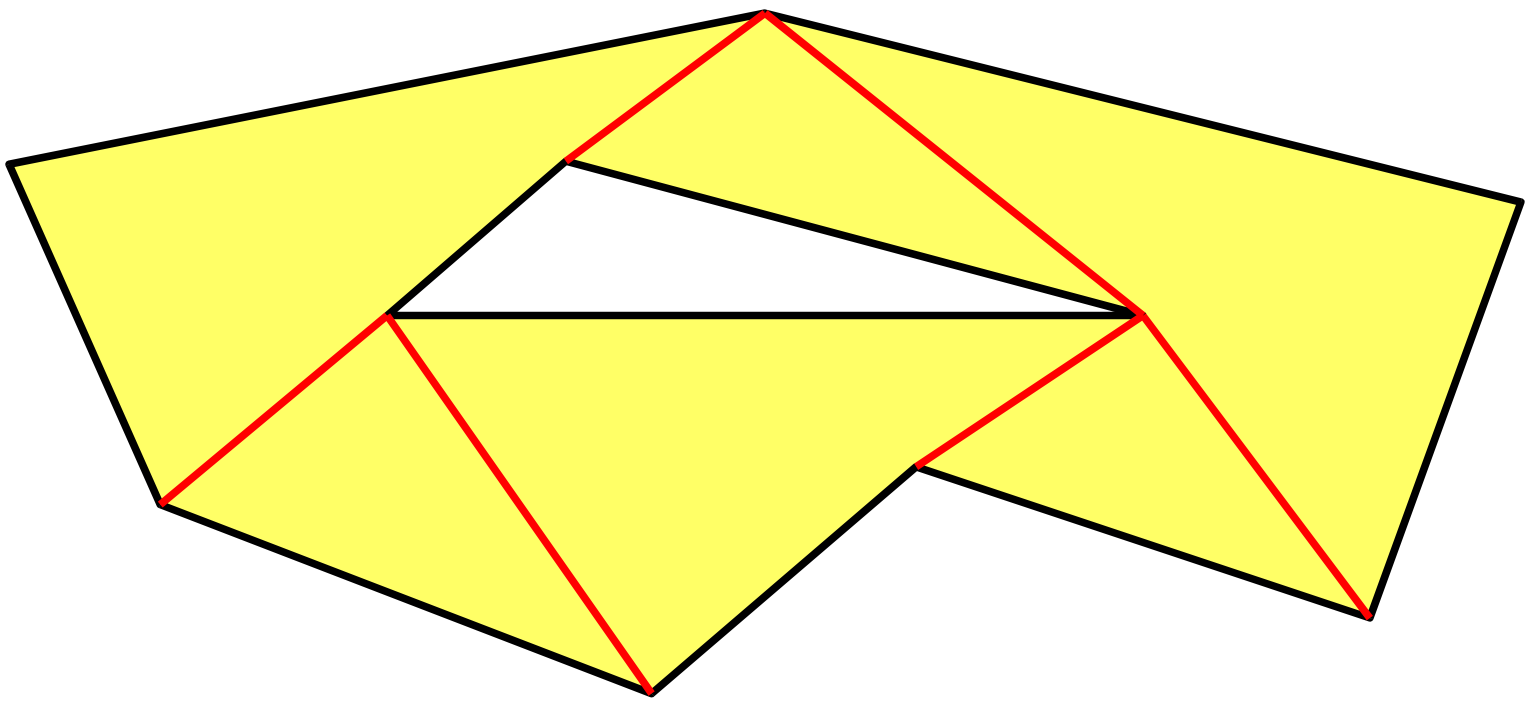
Benutzung der Trapezzerlegung

Wir gehen die verbleibenden Trapeze durch und fügen wo immer möglich Diagonalen ein.



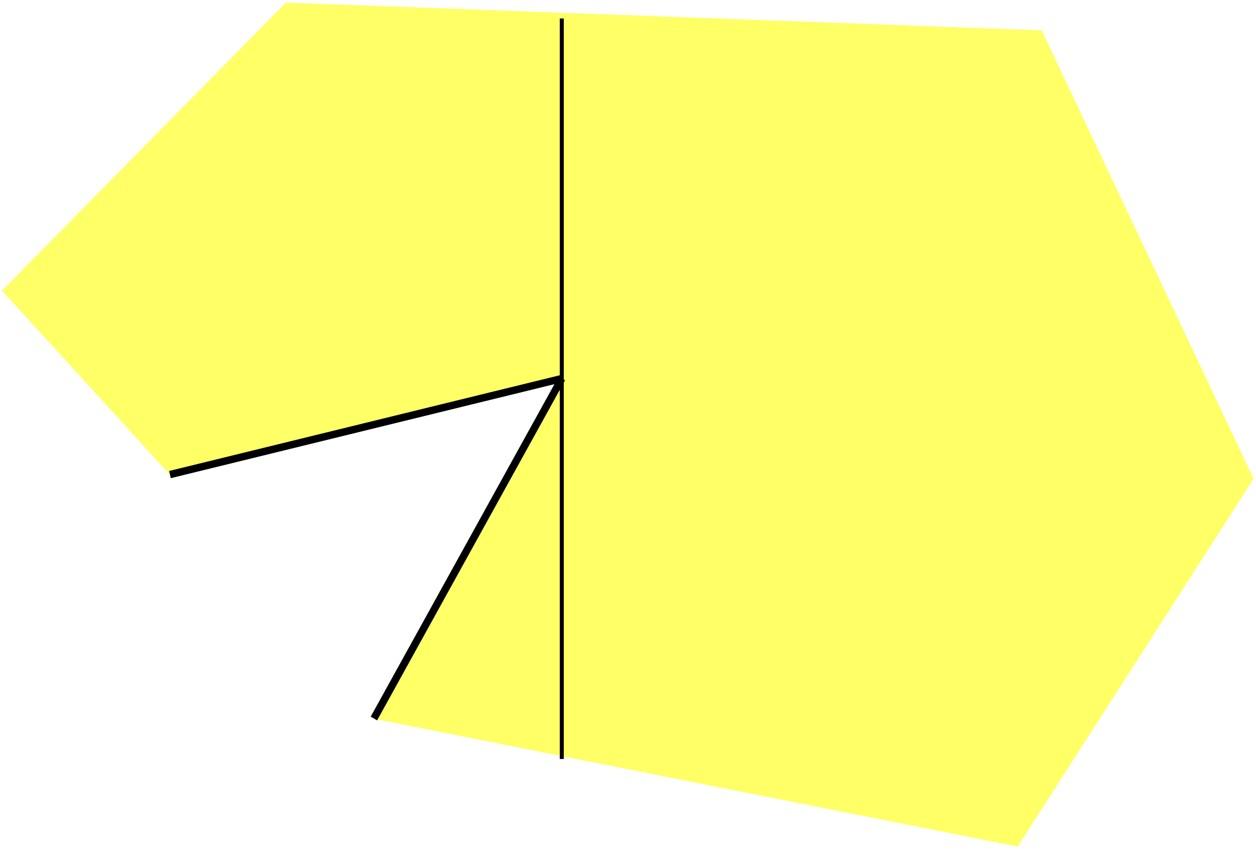
Beobachtung

Die so erhaltenen Teilpolygone haben scheinbar alle die Eigenschaft, dass der Schnitt mit jeder senkrechten Geraden eine Strecke, ein Punkt oder leer ist. Man nennt solche Polygone x-monoton.



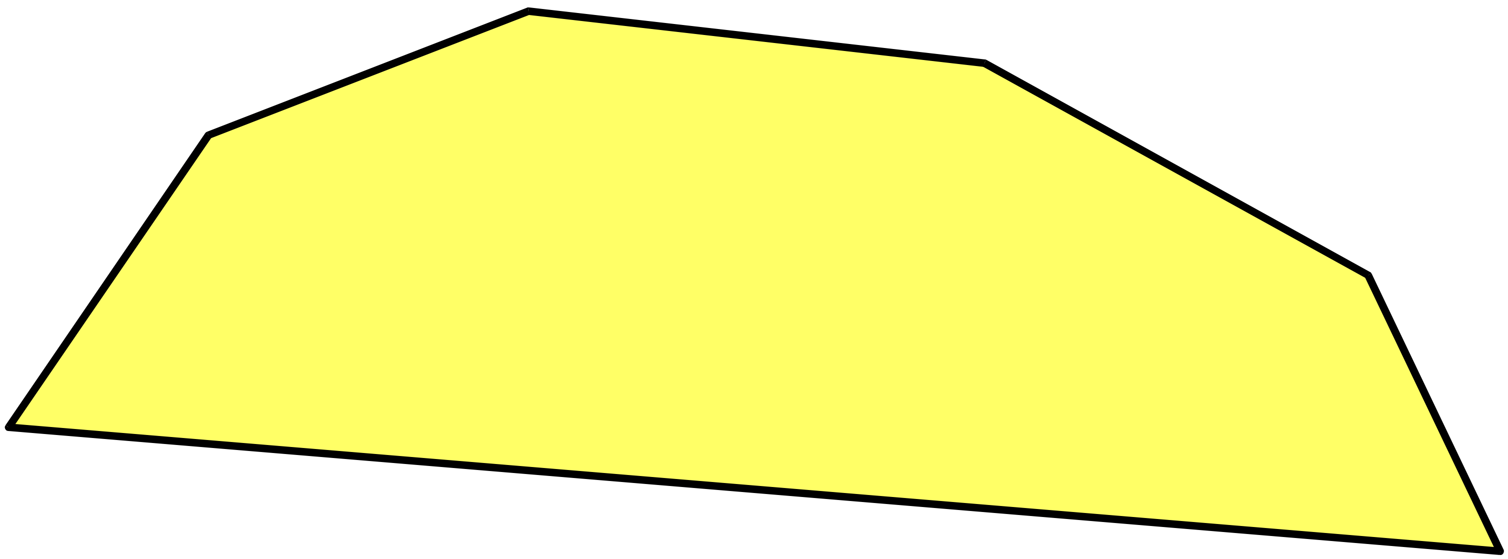
Beweis der beobachteten Eigenschaft

Angenommen ein Teilpolygon wäre nicht x-monoton.



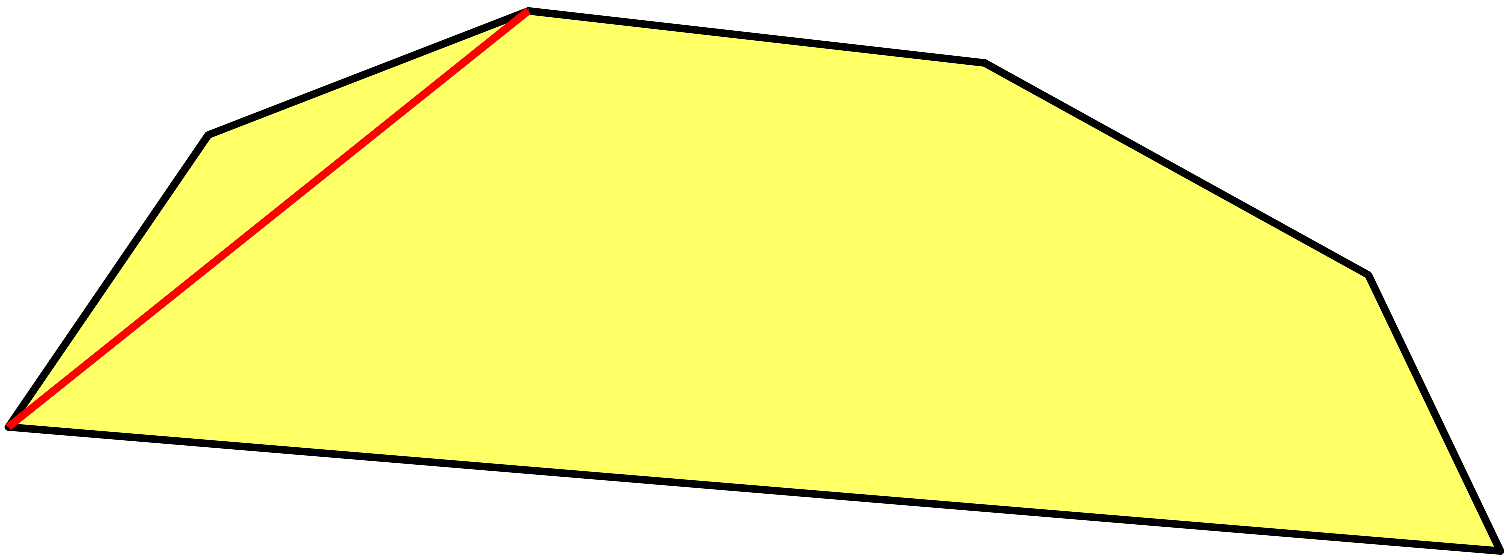
Triangulation von x-monotonen Polygonen

Wenn das Polygon schon x-monoton ist, dann liegt es nahe, die Ecken nach wachsenden x-Koordinaten abzuarbeiten.



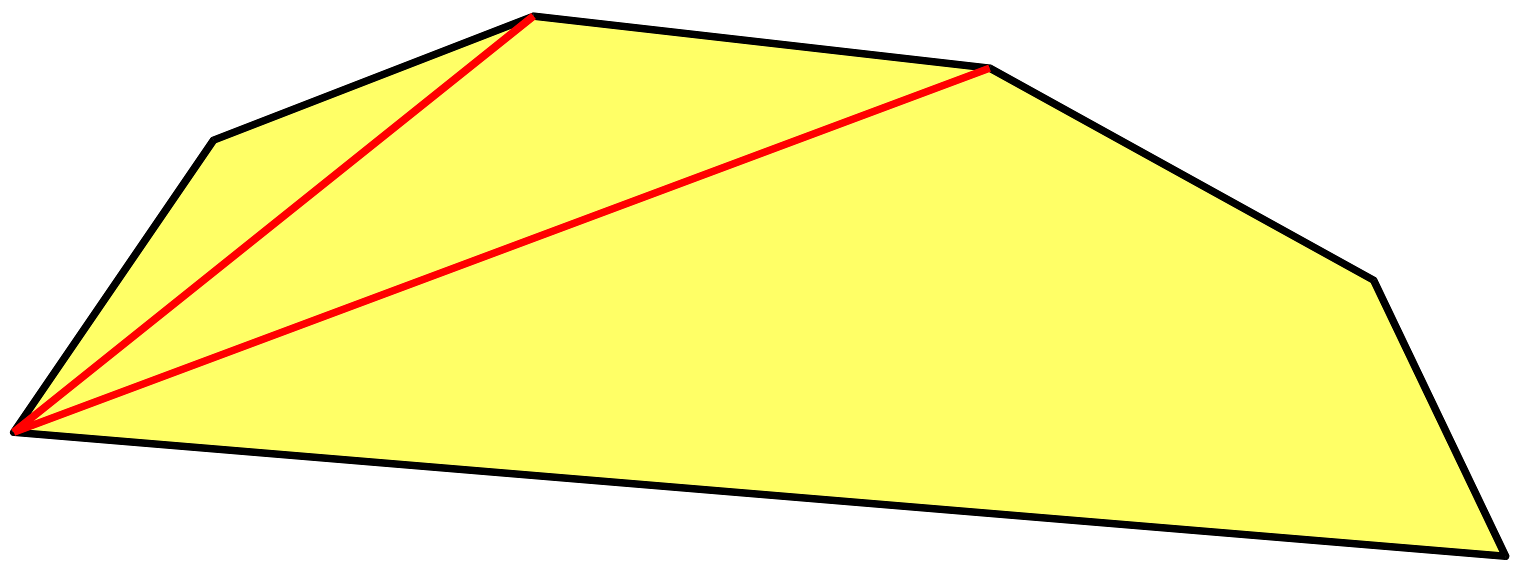
Triangulation von x-monotonen Polygonen

Wenn das Polygon schon x-monoton ist, dann liegt es nahe, die Ecken nach wachsenden x-Koordinaten abzuarbeiten.



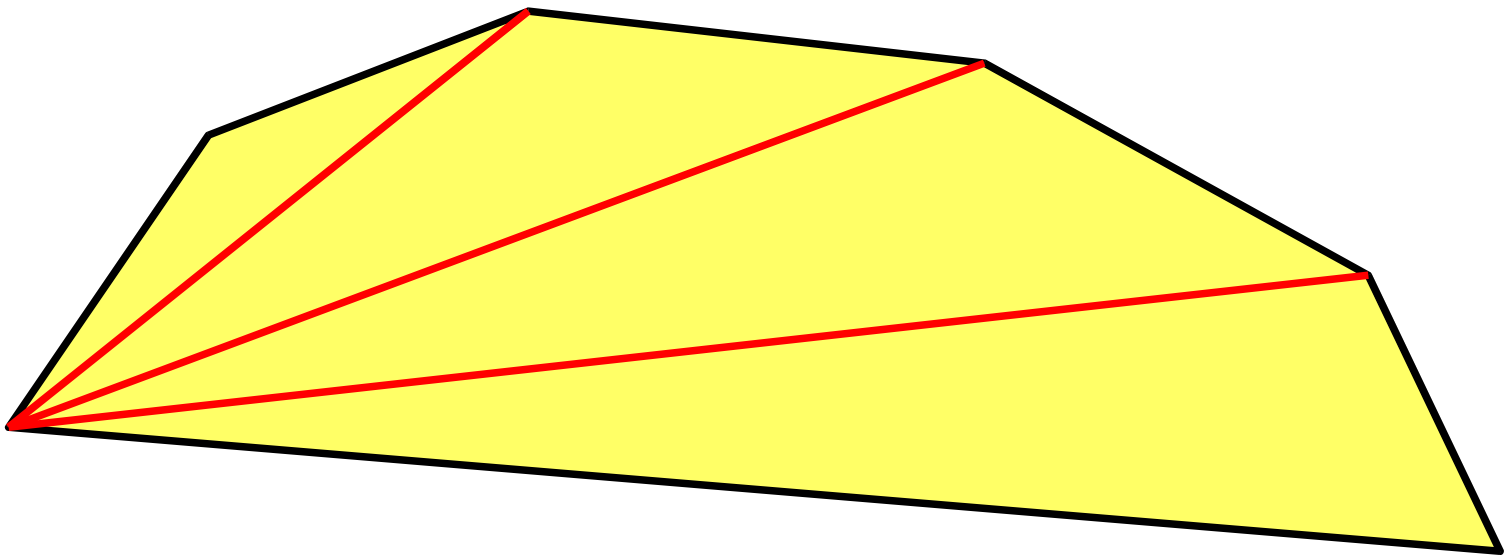
Triangulation von x-monotonen Polygonen

Wenn das Polygon schon x-monoton ist, dann liegt es nahe, die Ecken nach wachsenden x-Koordinaten abzuarbeiten.



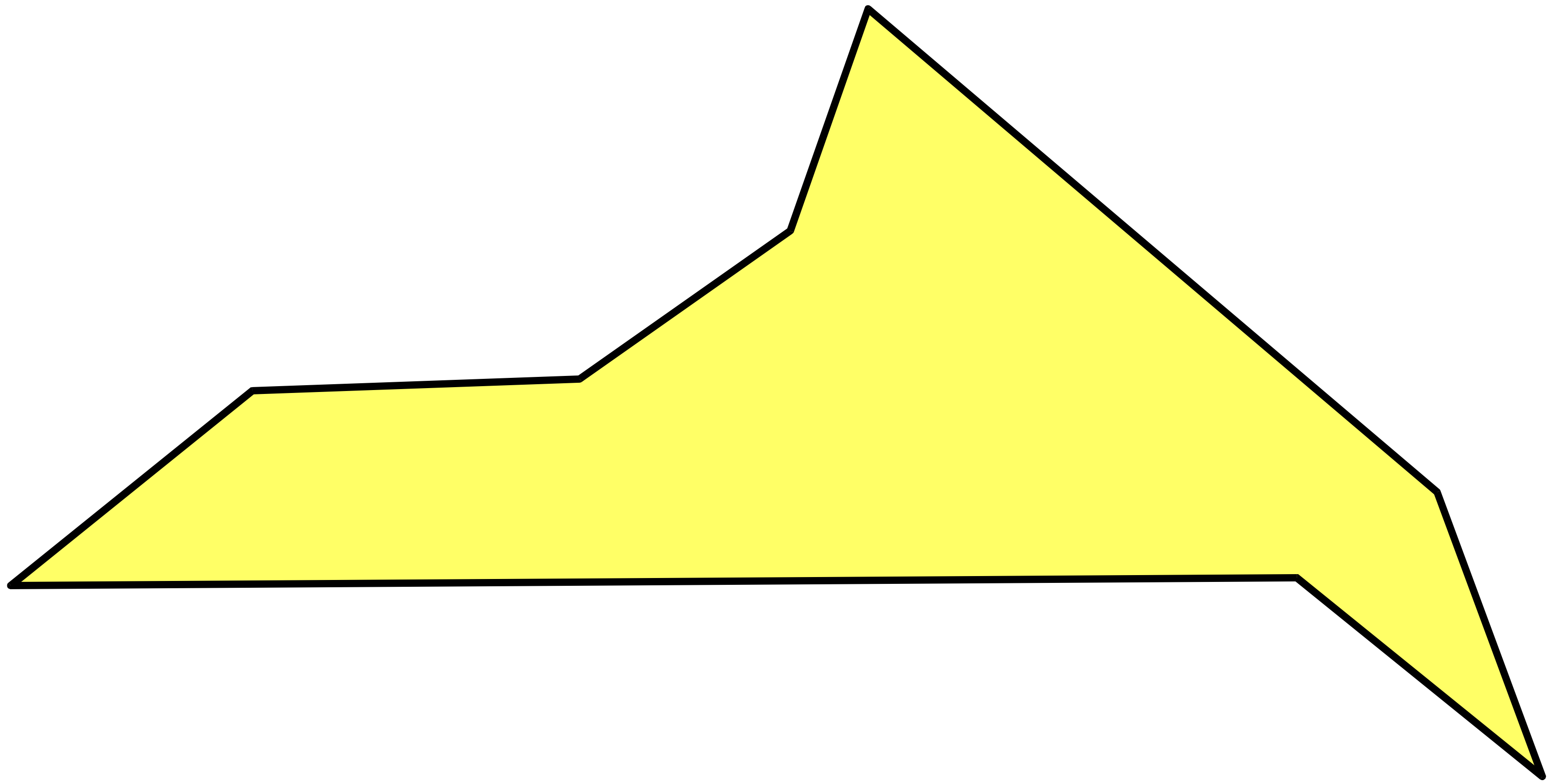
Triangulation von *x*-monotonen Polygonen

Wenn das Polygon schon x-monoton ist, dann liegt es nahe, die Ecken nach wachsenden x-Koordinaten abzuarbeiten.



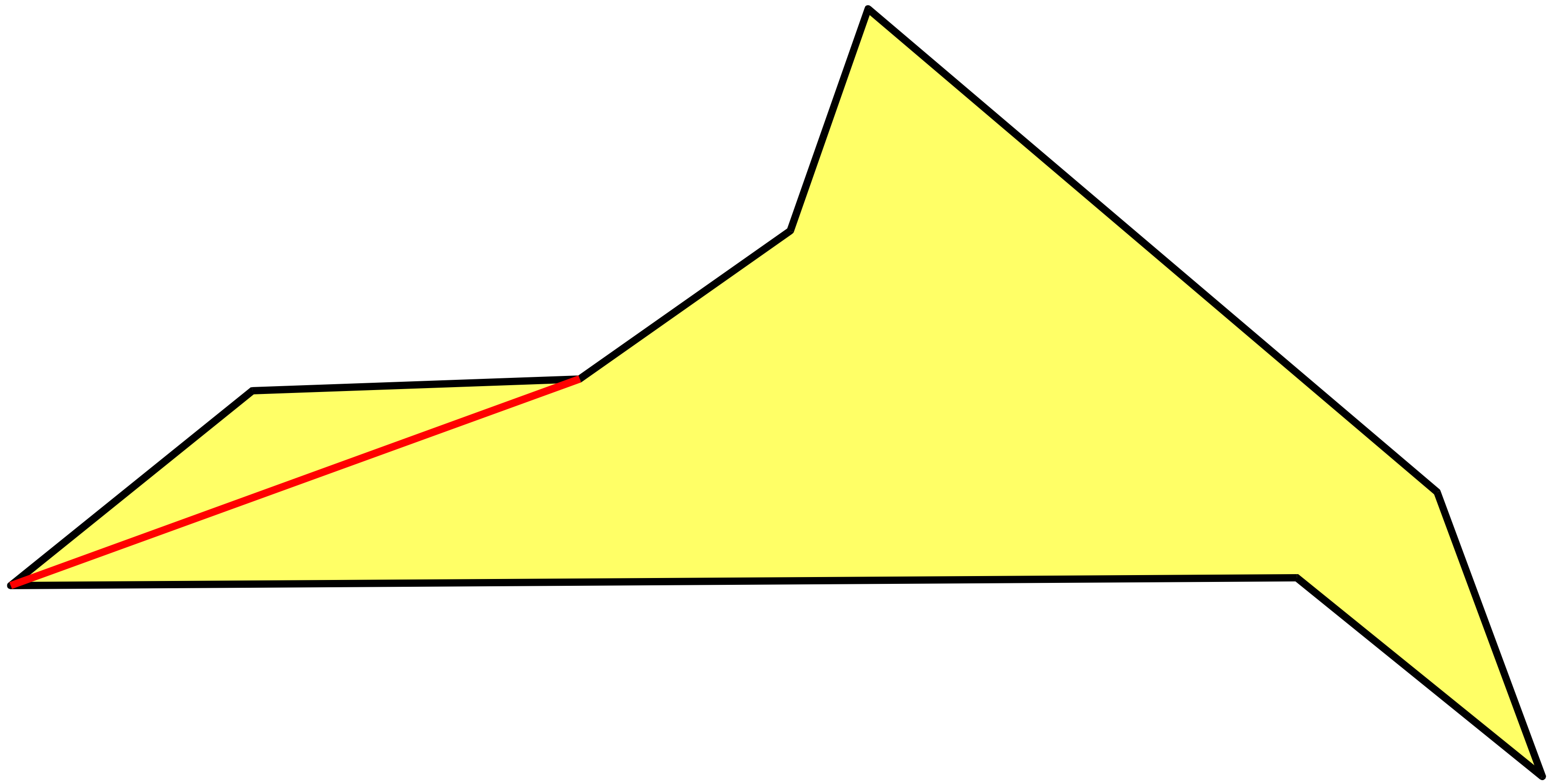
Triangulation von *x*-monotonen Polygonen

Leider lässt sich nicht jede Ecke sofort verarbeiten. Deshalb verwenden wir einen Stapel zur Verwaltung dieser aufgeschobenen Ecken.



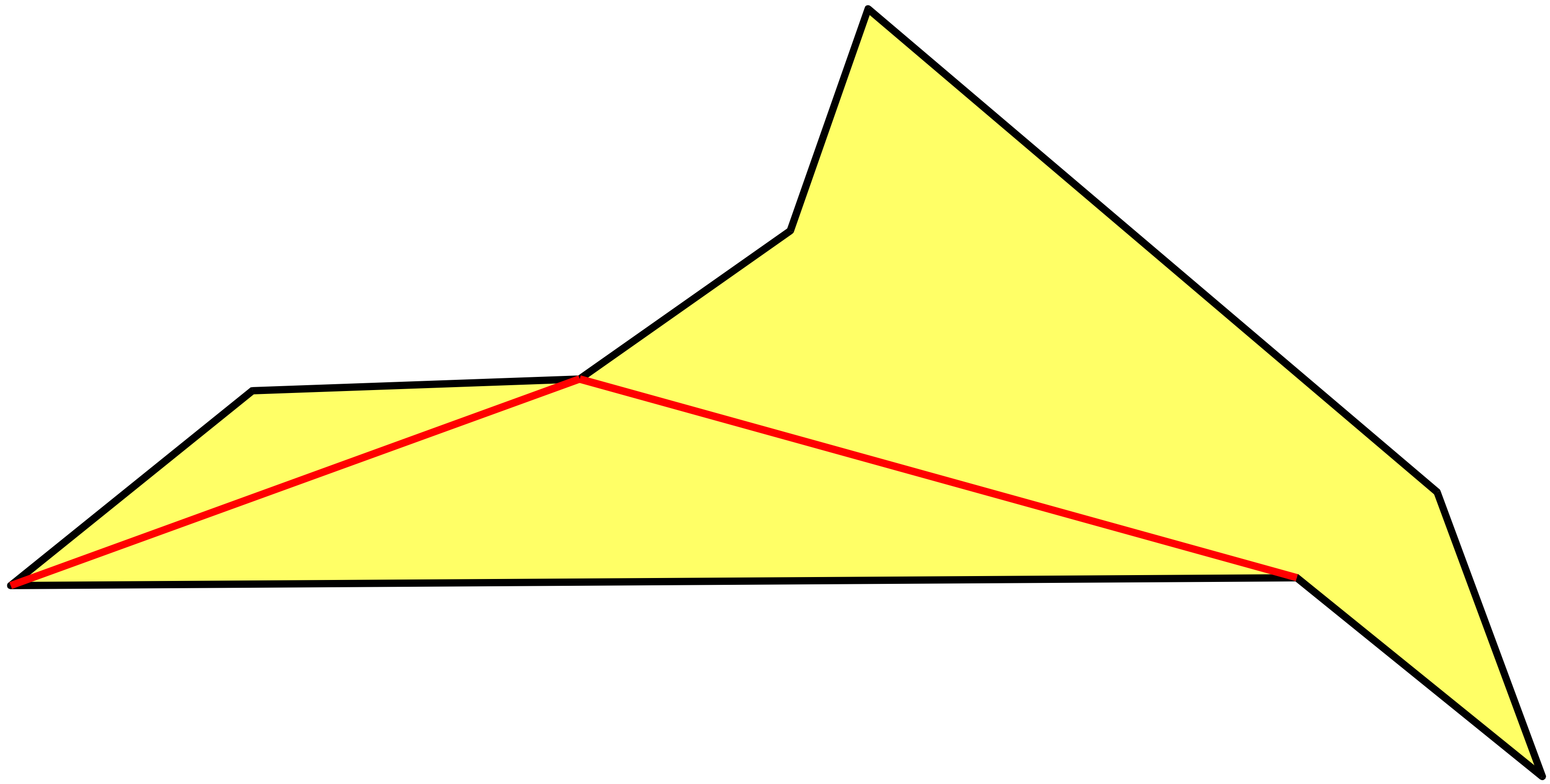
Triangulation von *x*-monotonen Polygonen

Leider lässt sich nicht jede Ecke sofort verarbeiten. Deshalb verwenden wir einen Stapel zur Verwaltung dieser aufgeschobenen Ecken.



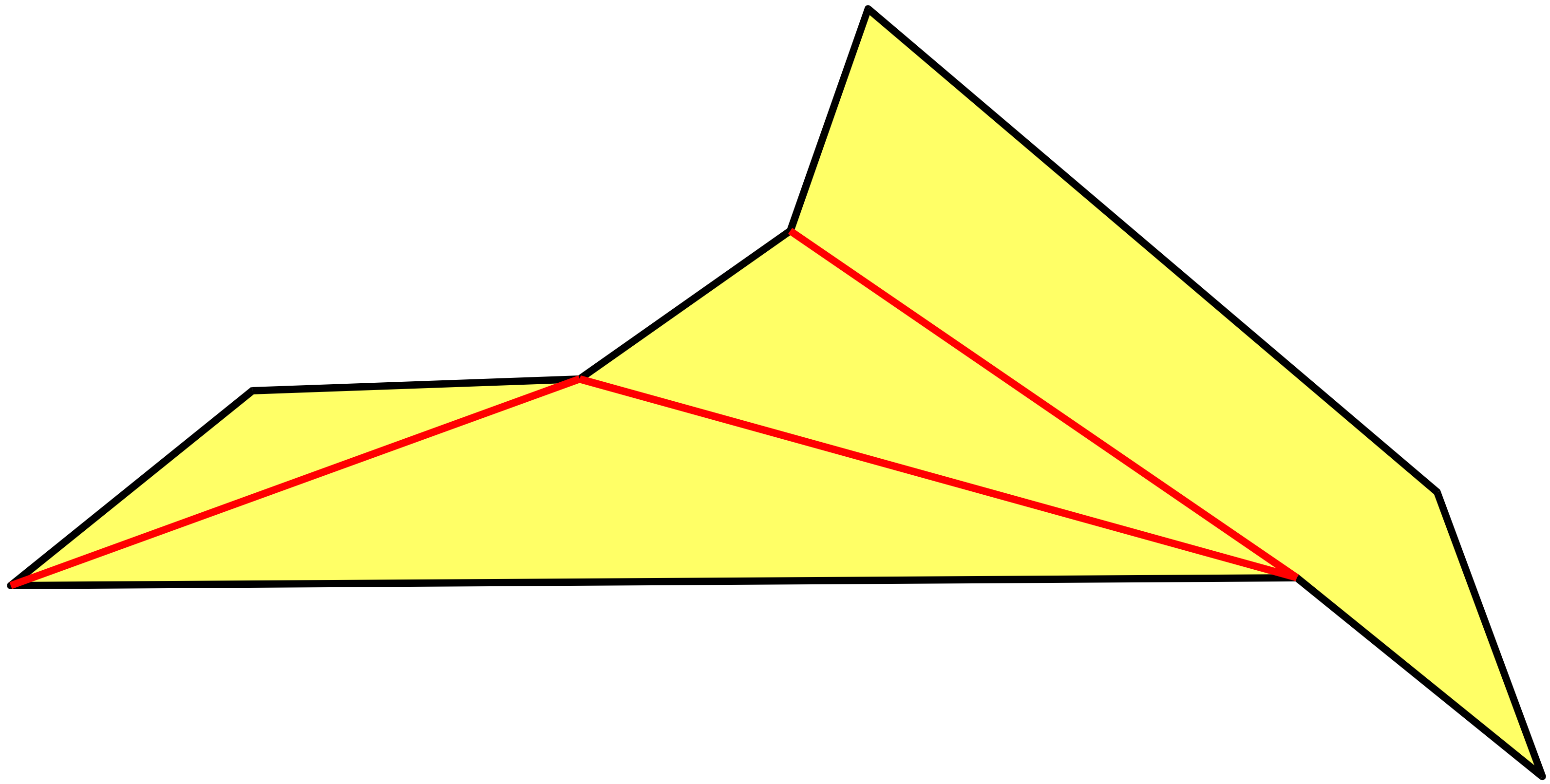
Triangulation von *x*-monotonen Polygonen

Leider lässt sich nicht jede Ecke sofort verarbeiten. Deshalb verwenden wir einen Stapel zur Verwaltung dieser aufgeschobenen Ecken.



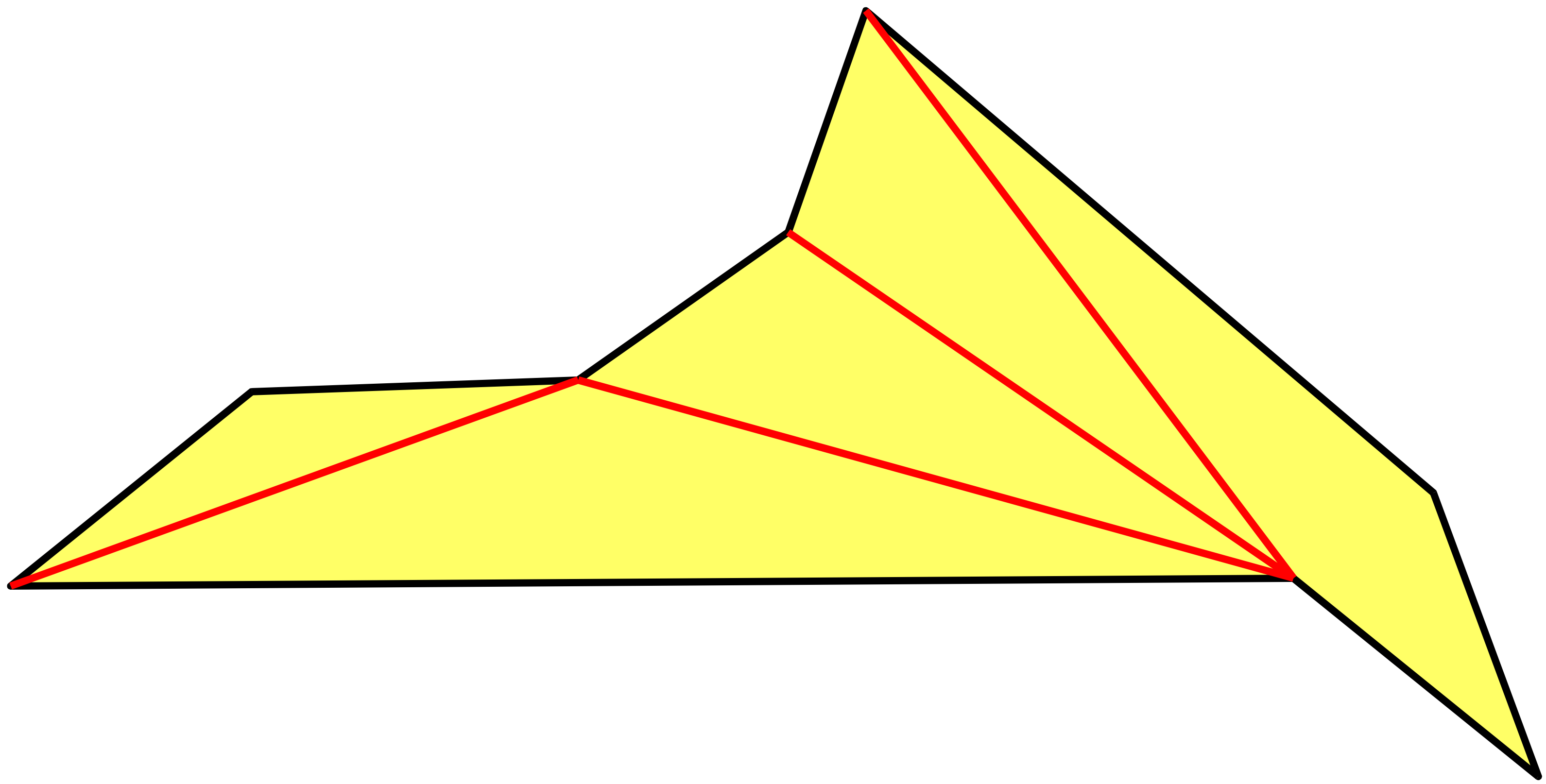
Triangulation von *x*-monotonen Polygonen

Leider lässt sich nicht jede Ecke sofort verarbeiten. Deshalb verwenden wir einen Stapel zur Verwaltung dieser aufgeschobenen Ecken.



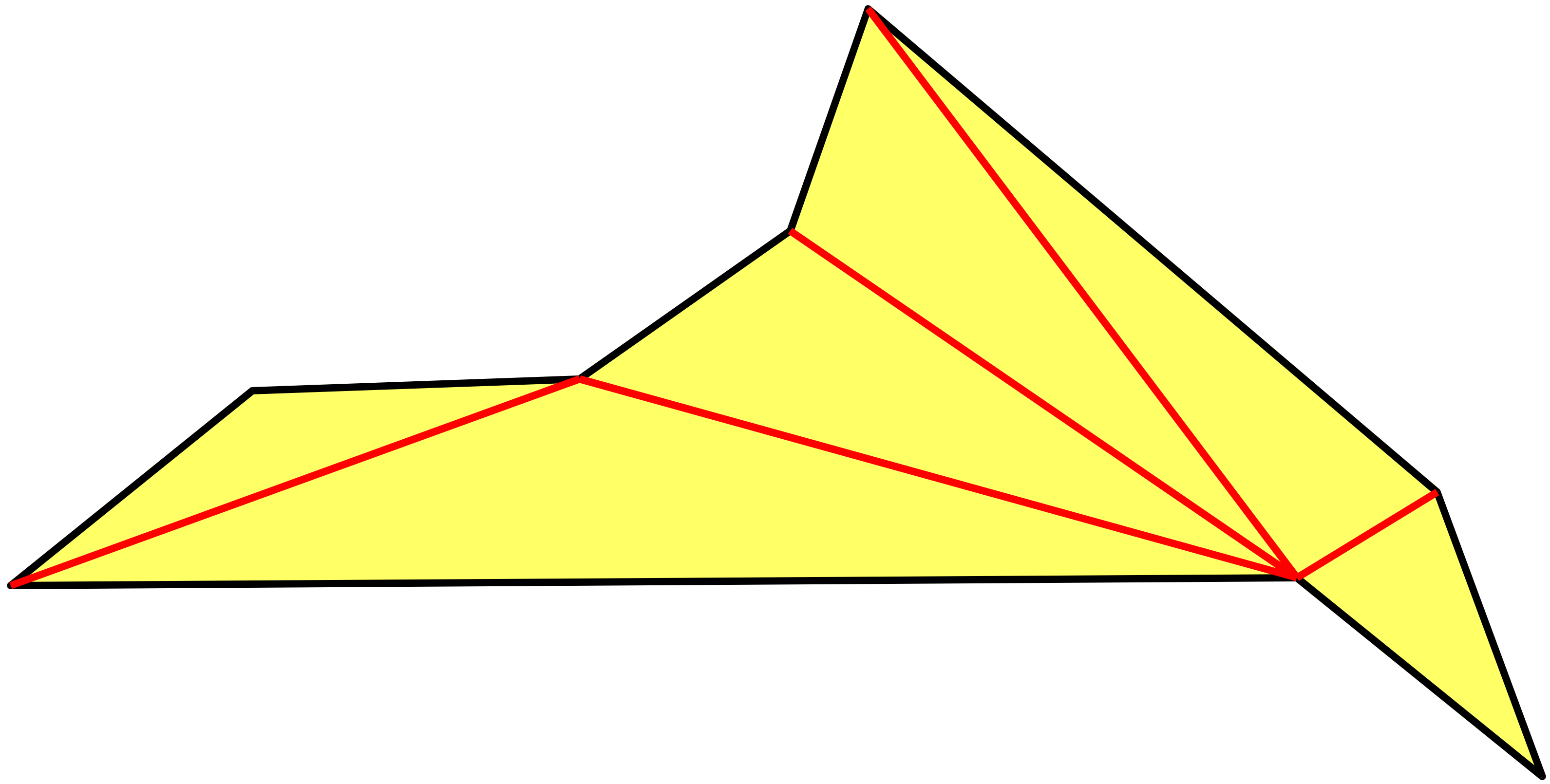
Triangulation von *x*-monotonen Polygonen

Leider lässt sich nicht jede Ecke sofort verarbeiten. Deshalb verwenden wir einen Stapel zur Verwaltung dieser aufgeschobenen Ecken.



Triangulation von *x*-monotonen Polygonen

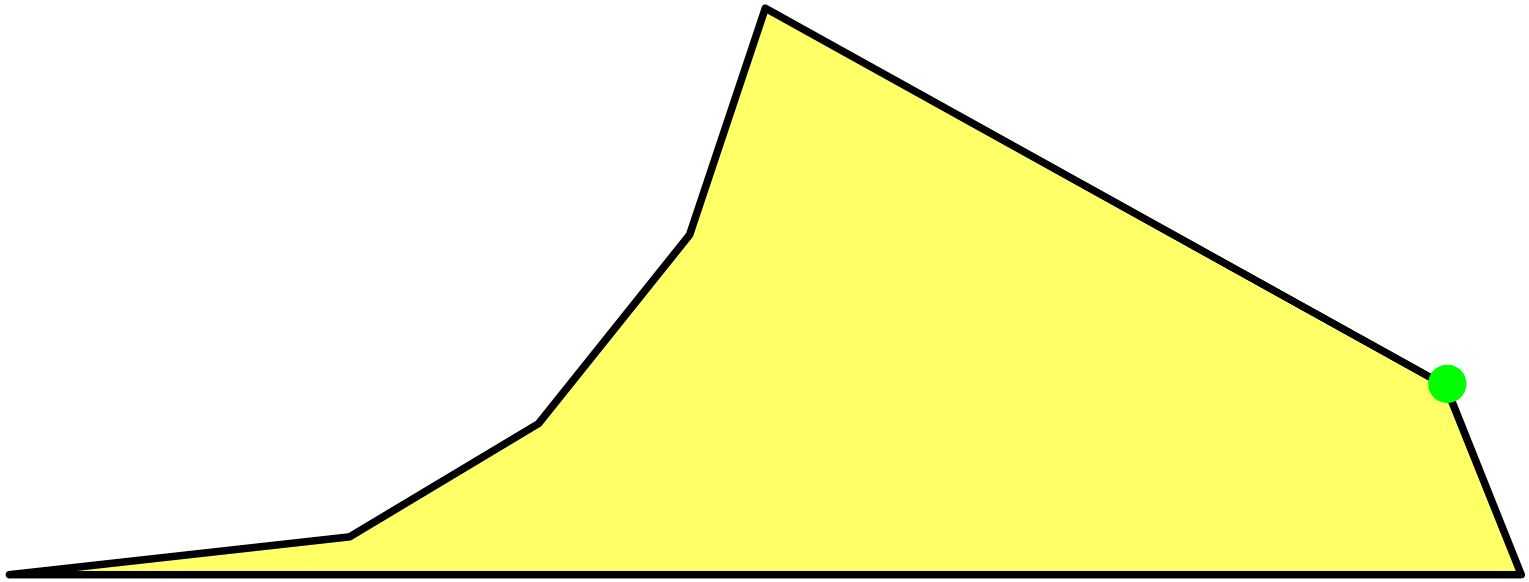
Leider lässt sich nicht jede Ecke sofort verarbeiten. Deshalb verwenden wir einen Stapel zur Verwaltung dieser aufgeschobenen Ecken.



Triangulation von *x*-monotonen Polygonen

Der Stapel wird am Anfang mit den ersten beiden Ecken initialisiert.

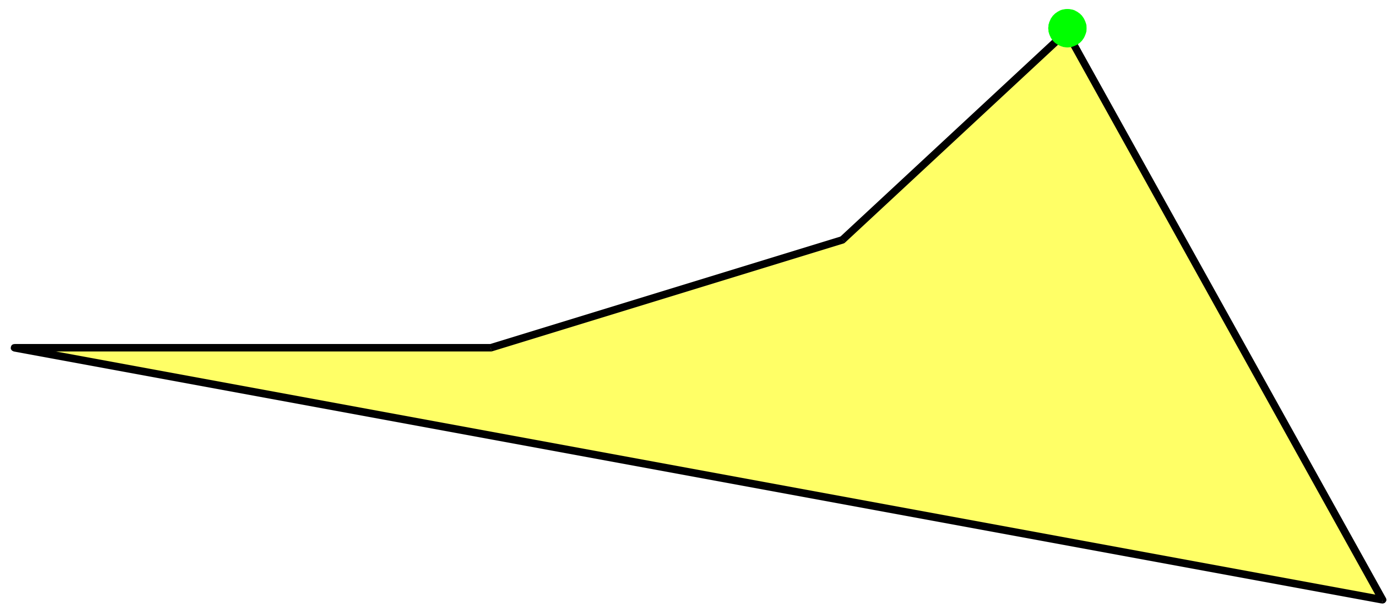
Es gilt auch später immer: Die Ecken im Stapel bilden eine reflexe Kette und das unterste Element ist die Ecke mit kleinster x-Koordinate im Restpolygon.



Triangulation von *x*-monotonen Polygonen

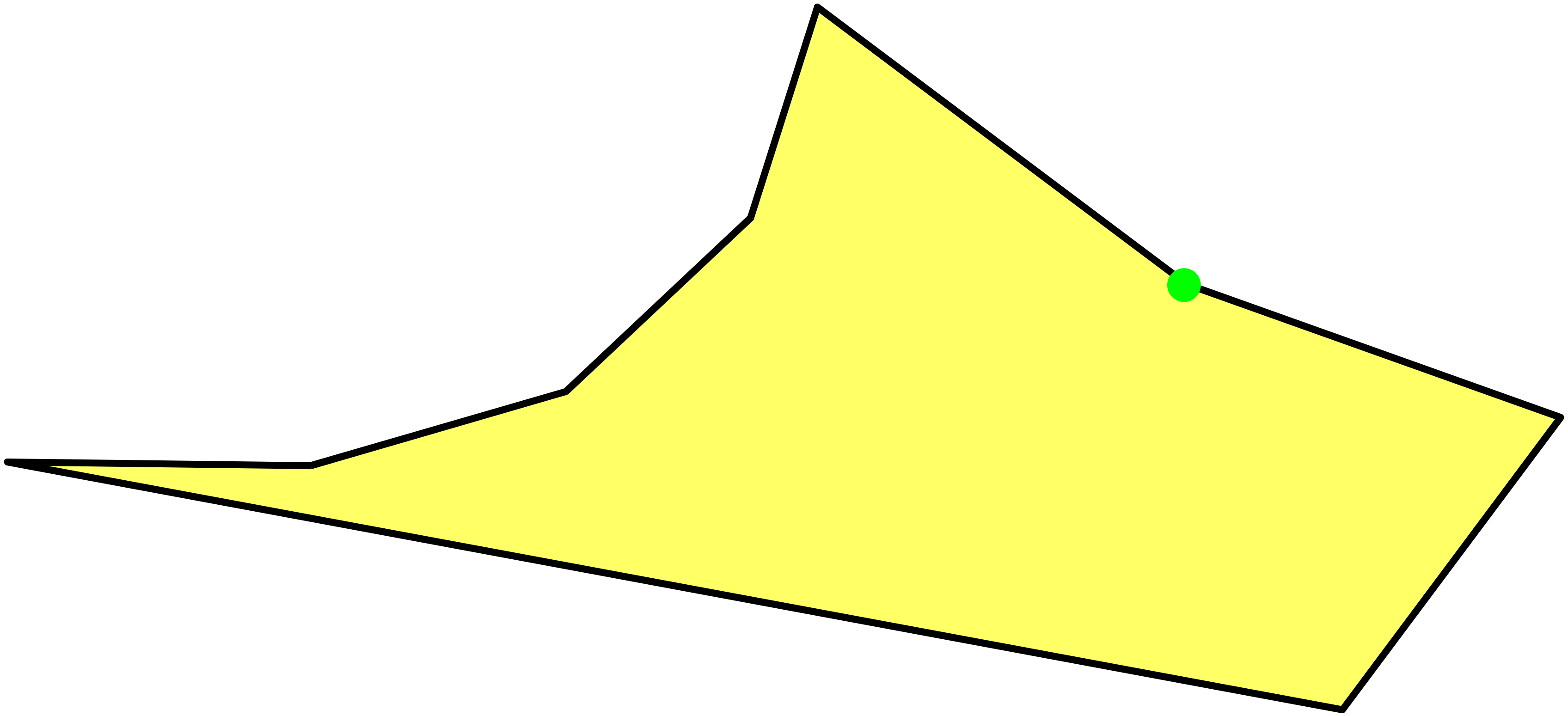
Bei der Bearbeitung der nächsten Ecke gibt es drei Fälle.

1.Fall: Die nächste Ecke verlängert die reflexe Kette.



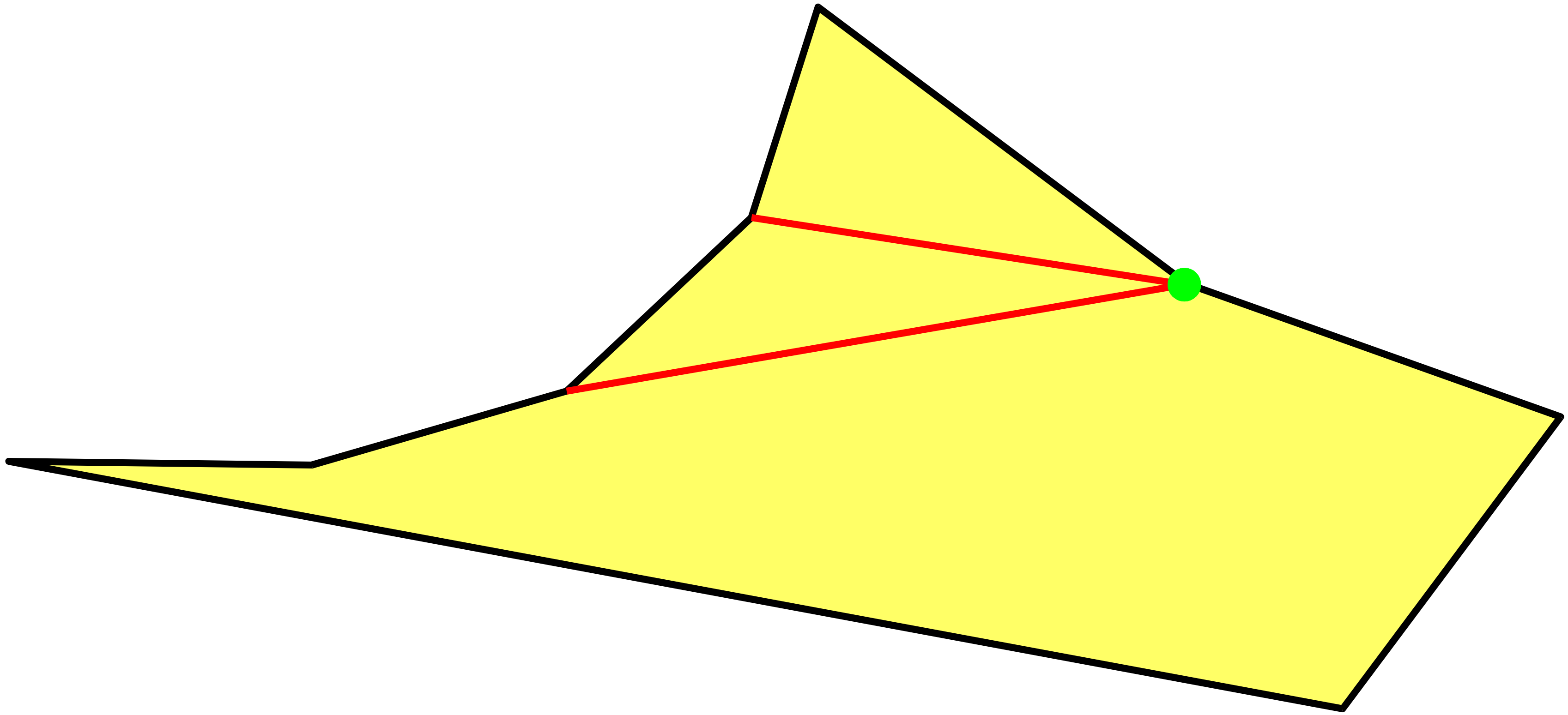
Triangulation von *x*-monotonen Polygonen

2.Fall: Die nächste Ecke verlängert die reflexe Kette nicht und liegt auf der gleichen Seite wie die reflexe Kette.



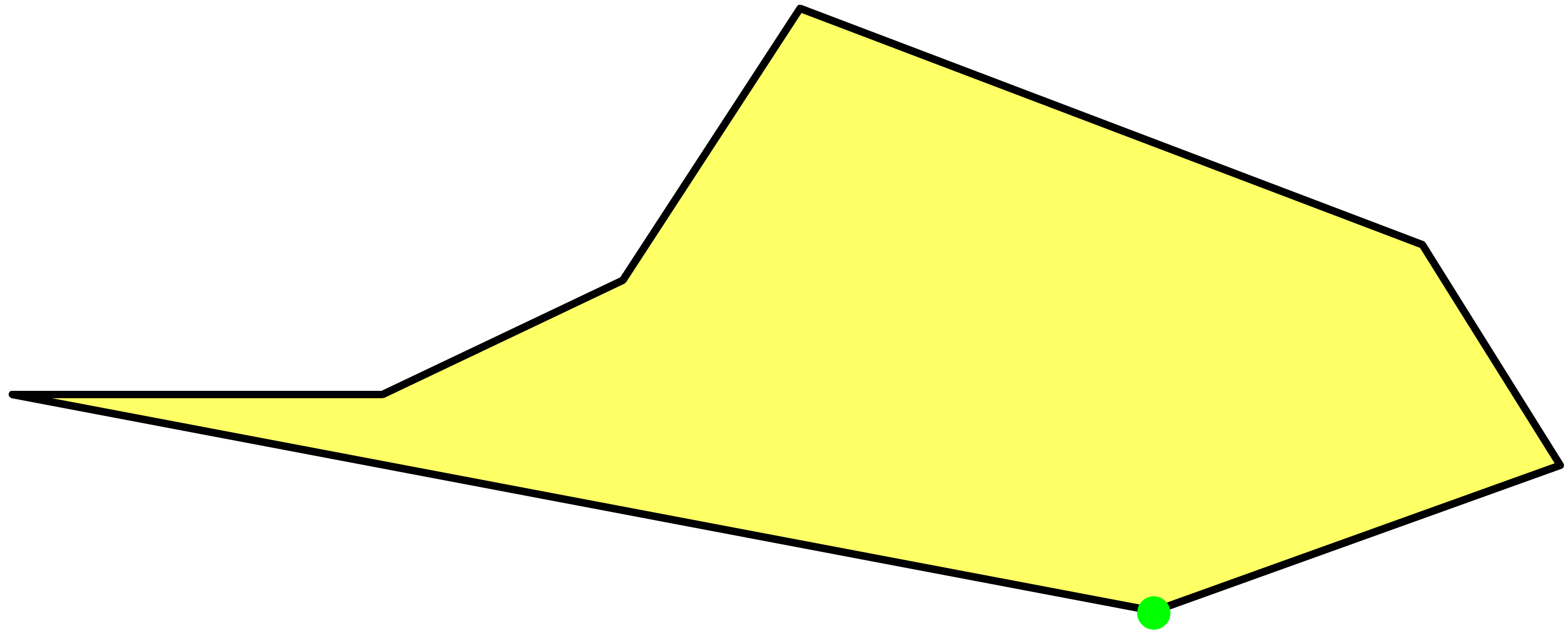
Triangulation von *x*-monotonen Polygonen

Die reflexe Kette wird entsprechend gekürzt und die bearbeitete Ecke anschließend auf den Stapel gelegt.



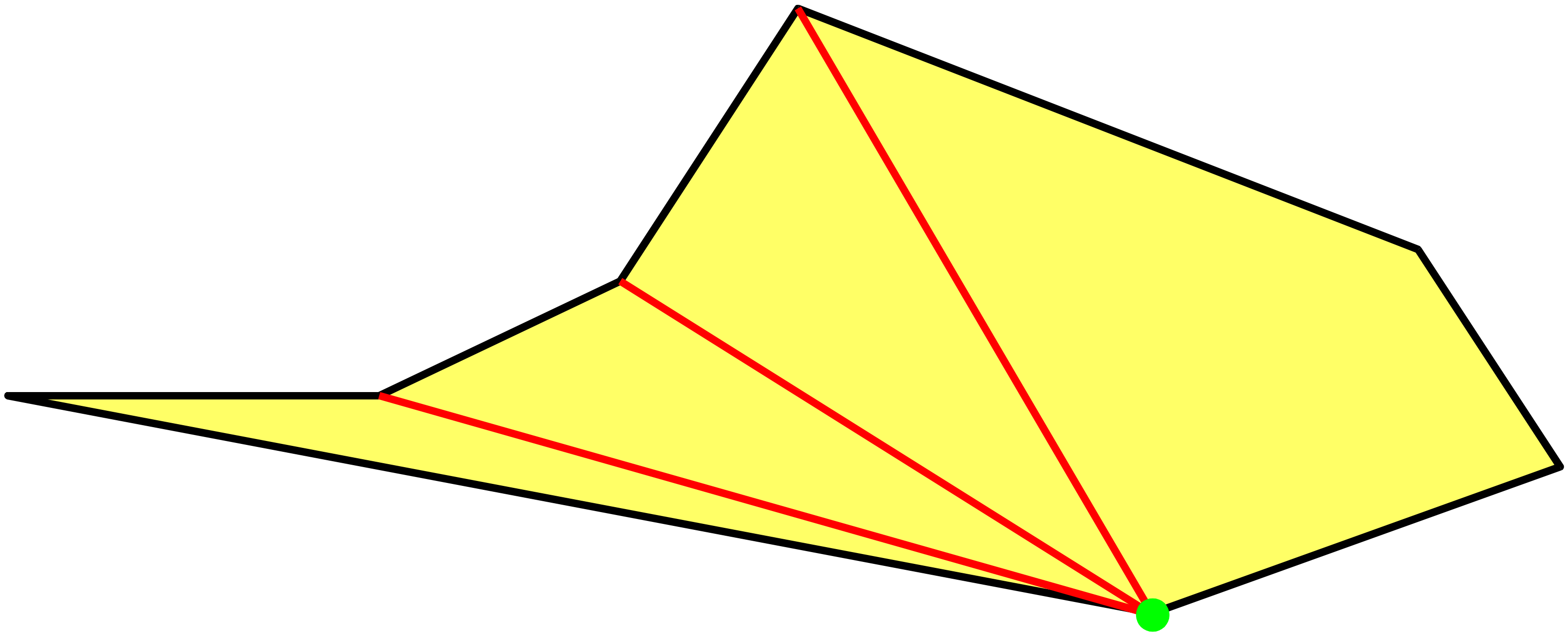
Triangulation von *x*-monotonen Polygonen

3.Fall: Die nächste Ecke verlängert die reflexe Kette nicht und liegt nicht auf der gleichen Seite wie die reflexe Kette.



Triangulation von *x*-monotonen Polygonen

Alle Ecken auf dem Stapel werden abgearbeitet. Dann wird der Stapel mit den beiden ersten Ecken im Restpolygon initialisiert.



Analyse der Laufzeit

Berechnung der Trapezzerlegung: $O(n \log n)$

Entfernen der äußeren Trapeze: $O(n)$

Zerlegung in x-monotone Polygone: $O(n)$

Triangulation der x-monotonen Polygone: $O(n)$

Gesamtlaufzeit: $O(n \log n)$