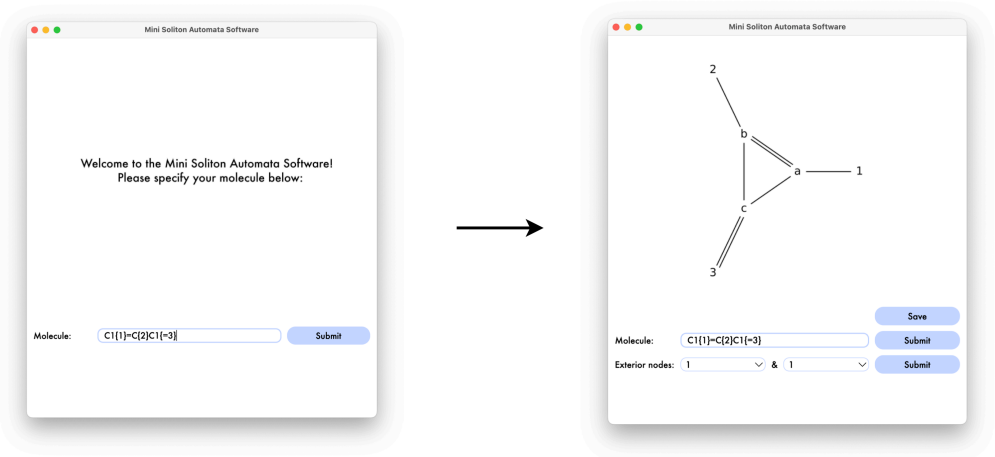
Informationen zur Mini Soliton Automata Software

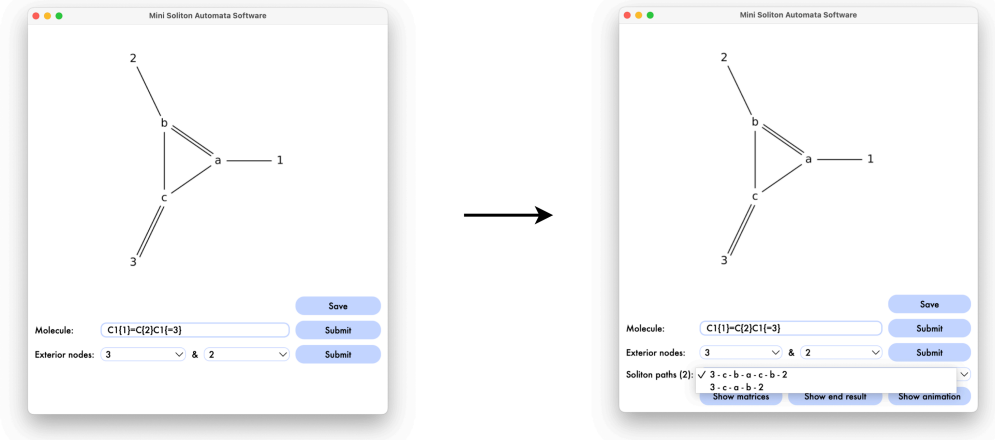
Mithilfe der Mini Soliton Automata Software lassen sich alle Solitonpfade zwischen zwei externen Knoten in einem Solitongraphen berechnen. Ihre graphische Benutzeroberfläche ermöglicht verschiedene Eingaben und zeigt Visualisierungen des Solitongraphs und der gefundenen Solitonpfade. Unter https://github.com/schulz-helena/soliton-implementation befindet sich das Repository zur Software.

Bedienungsanleitung:

Nach dem Starten der Software ist zunächst im verfügbaren Eingabefeld das Molekül anzugeben, in welchem nach Solitonpfaden gesucht werden soll. Die Spezifizierung eines Moleküls folgt einer bestimmten Syntax (siehe Abschnitt „Input-Syntax für die Spezifizierung eines Moleküls“). Mit „Submit“ kann die Eingabe bestätigt werden. Nun wird im oberen Bereich der Benutzeroberfläche der entsprechende Solitongraph angezeigt. In der Visualisierung sind innere Knoten als Buchstaben und externe Knoten als Zahlen gekennzeichnet. Mit „Save“ kann der Graph als PNG-, JPG- oder JPEG-Datei gespeichert werden.

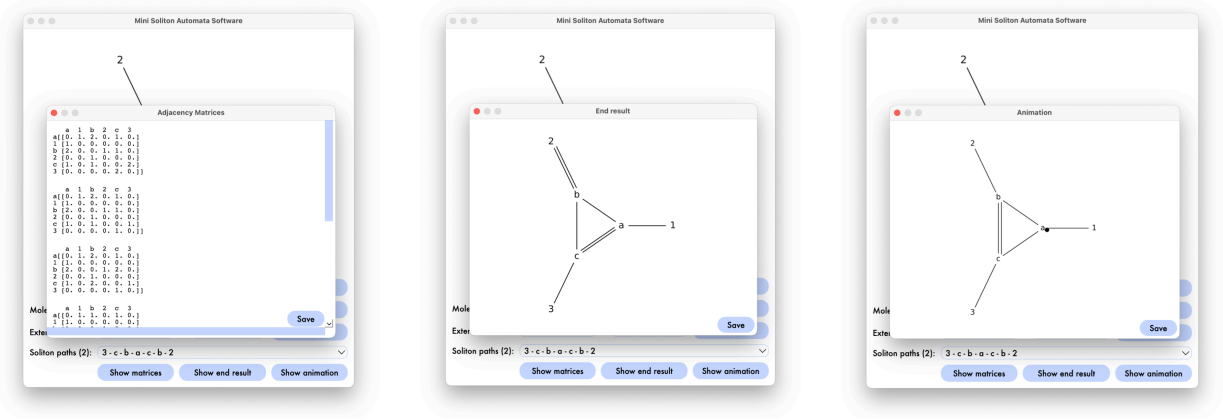


Als nächstes können aus allen externen Knoten des Solitongraphs zwei externe Knoten ausgewählt werden. Im ersten ausgewählten Knoten betritt das Soliton den Solitongraphen und im zweiten verlässt es ihn wieder. Ist diese Eingabe mit dem „Submit“-Button direkt hinter der Knotenauswahl bestätigt worden, werden alle möglichen Solitonpfade zwischen den beiden externen Knoten berechnet.



Nun kann einer der berechneten Pfade ausgewählt werden. Mit „Show matrices“ öffnet sich ein Fenster mit den Ajazenzmatrizen des Solitongraphs zu jedem Zeitpunkt des Durchlaufens des Solitonpfades. Mit „Save“ kann ein Textdokument heruntergeladen werden, was diese

Adjazenzmatrizen, das spezifizierte Molekül und den ausgewählten Solitonpfad enthält. „Show end result“ öffnet ein Fenster, in dem der resultierende Solitongraph (nachdem das Soliton den Pfad durchlaufen hat) angezeigt wird. Auch diese Visualisierung ist herunterladbar. Ein Fenster, was eine Animation zeigt, in dem das Soliton den ausgewählten Pfad durchläuft, lässt sich mit „Show Animation“ öffnen. Hier gibt es die Möglichkeit, diese Animation als GIF abzuspeichern.



Input-Syntax für die Spezifizierung eines Moleküls:

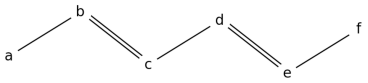
Die Input-Syntax basiert auf dem Simplified Molecular Input Line Entry Specification (SMILES) Strukturcode, mit dem Moleküle als Strings dargestellt werden können. Diese SMILES-Repräsentation wird um eine Regel zum Spezifizieren von externen Knoten erweitert.

Kohlenstoffatome werden als *C* dargestellt.

a
C ⟶ a

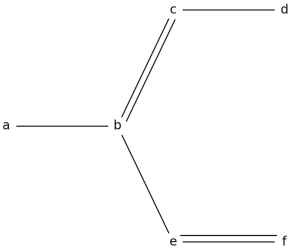
Einfachbindungen werden durch ein - oder gar kein Zeichen markiert (*C-C* entspricht *CC*). Doppelbindungen werden durch ein = markiert.

a b c d e f
CC=CC=CC
C-C=C-C=C-C ⟶

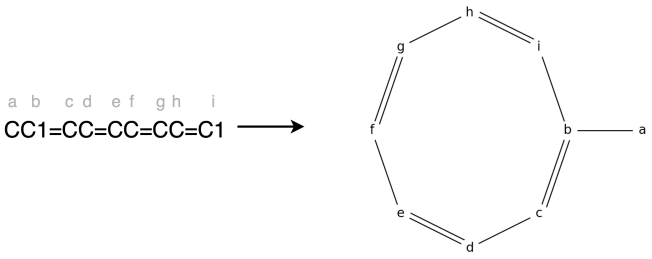


Gibt es eine Abzweigung, also führen aus einem Atom insgesamt drei Bindungen heraus, so muss diese Abzweigung (eine der Bindungen und alle ihr nachfolgenden Atome und Bindungen) in runde Klammern geschrieben werden. Die andere Bindung und alle ihr nachfolgenden Atome und Bindungen folgen nach den runden Klammern.

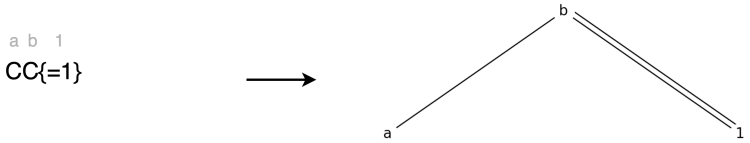
a b c d e f
CC(=CC)C=C ⟶



Bei einem Ring werden die Atome an der Stelle, an der der Ring geschlossen wird, mit der gleichen Nummer markiert (z.B. *C1* und *C1*). Zwischen diesen beiden Atomen wird eine Einfachbindung gezogen.

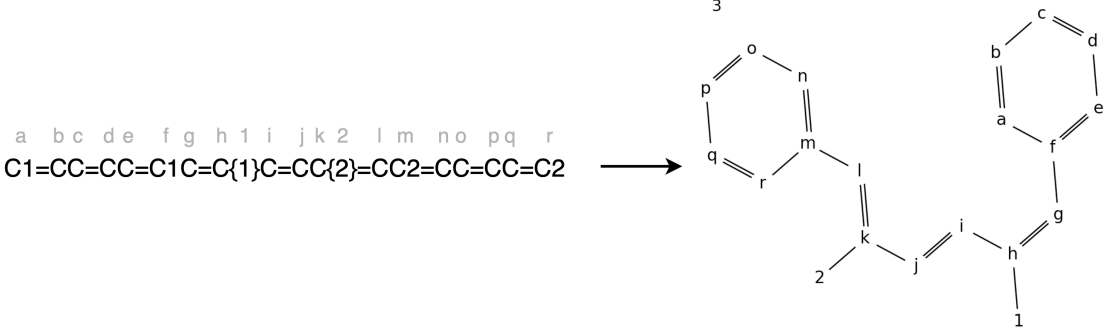


Externe Knoten werden als eine Bindung und eine Zahl in einer geschweiften Klammer dargestellt.



Wichtig bei der Benutzung von Klammern (*()* bei einer Abzweigung oder *{}* bei einem externen Knoten) ist, dass die Bindung zwischen dem Atom vor der öffnenden Klammer und dem Atom hinter der öffnenden Klammer immer in die Klammer geschrieben wird (z.B. *C(=C)* statt *C=(C)*).

Beispiele für Solitongraphen:



Implementierung:

Programmiersprache:

Die Mini Soliton Automata Software ist komplett in Python geschrieben.

Gewählte Bedingung für Pfadberechnung:

Der Algorithmus zur Berechnung der Solitonpfade benutzt die „Vergangenheitsbedingung“. Das heißt, jedes Mal, wenn entschieden wird, welcher Knoten als nächstes betreten werden kann, wird die zuletzt traversierte Kante betrachtet. Nur wenn diese ein anderes Gewicht hat, als die Kante zu einem möglichen nächsten Knoten, kann dieser Knoten ausgewählt werden.

Benutzte Frameworks:

- *rdkit*: Wandelt den SMILES-String in eine Repräsentation eines Moleküls um. Dieses Molekül fungiert als eine Art Zwischendarstellung und wird später in einen Graphen umgewandelt. Berechnet Positionen für Atome und Bindungen, die einer chemischen Darstellung eines Moleküls entsprechen, weshalb diese Positionen auch für die Darstellung als Solitongraph genutzt werden.
- *pysmiles*: Berechnet die exakten Bindungstypen jeder Bindung.
- *networkx*: Plottet den Graphen.
- *matplotlib*: Plottet auf die networkx-Darstellung des Graphs für jede Doppelbindung eine zweite Kante und einen schwarzen Punkt als Repräsentation des Solitons.
- *numpy*: Realisiert die Adjazenzmatrizen des Solitongraphs.
- *Pillow*: Wandelt Visualisierungen und Animationen des Solitongraphs in Images um und gibt damit die Möglichkeit, diese im Arbeitsspeicher zu halten, statt sie lokal abzuspeichern.
- *PyQt5*: Realisiert die graphische Benutzeroberfläche.

Installation und Starten der Software:

Um die Software benutzen zu können, wird Python benötigt. Auf dem Zielrechner, bzw. der Zielumgebung sollte Python 3.9 oder höher installiert sein.
Auch der Python Package Installer pip wird benötigt.
Sind diese Voraussetzungen erfüllt, kann die Mini Soliton Automata Software mittels pip von GitHub heruntergeladen werden

Mit dem Kommandozeilenbefehl
`pip install git+https://github.com/schulz-helena/soliton-implementation`
wird die Software als Python Package heruntergeladen.

Nun kann sie mittels
`mini-soliton-automata-software`
gestartet werden.