## Entwicklung der „delete to preserve“ Implementierung:

* Über die Liste der zu löschenden Knoten iterieren.
  + Hinzufügen entsprechender Knoten zur RHS und erstellen der entsprechenden Mappings
* Über die Liste der zu löschenden Kanten iterieren.
  + Hinzufügen der entsprechenden Kante zur RHS.

**Erledigt!**

**TODO:** Wie sollte Softwaretechnik die Methode Robust gemacht werden und Error Handling betreiben?

* Bisher arbeitet die Methode auf Grundlage von im Dateisystem persistierten Regeln.
  + Wofür sollte die Methode bereitstehen? Dateisystem? Geladene Regeln?
* Soll etwas der Art auch für die UI eingesetzt werden? Eigentlich sollte nur das Analysiert werden, was der Nutzer an Regeln als Eingabe liefert. Eine Analyse auf Grundlage anderer Regeln als der Nutzer bereitstellt ist nicht intuitiv.

## Entwicklung des *Wrapper* um die *delete-use*-Analyse auch für *produce-use* zu nutzen:

## Anpassen des „Runner“ für allgemeineres Verhalten!

Vorhalten aller CPA, essCPA und coC Ergebnisse!

## Programmatisches erstellen der „use“ Regeln für delete-use overapproximation gegenüber ursprünglichen delete-delete Konflikten.

Die CPA führt zu keiner over approximation.

Bei unserem Ansatz kommt es zu einer over approximation.  
Ursache dafür ist: für das durch CoC gemeldete Ergebnis wäre die zweite Regel eigentlich gar nicht anwendbar, da es dann zu einer dangling edge kommt.

Eigenes kleines Beispiel:

Gab/ gibt es dazu nicht auch ein Beispiel von Gabi?

Wie lässt sich dies „vermessen“?

Daniels input: nicht einfach nur absolute Summe der gemeldeten Konflikte der ganzen Regelmenge gegenüberstellen, sondern differenzierter betrachten. Wie viele der in Konflikt stehenden Paare sind betroffen? Wie stark ist dort die Überapproximation?

Generelle Überlegung: Ist es nicht mit einer Art *postprocessing* möglich zu prüfen, ob die Regel auf dem gefundenen / gemeldeten *Overlap* anwendbar ist?

Weitere TODOs:

* Tests schreiben!
* In Doku schreiben, dass nur normale Knoten behandelt werden.
  + Keine MultiRules
  + Keien Attribute