

Test test test

- Notiz: Max Wert von Initialisierung wird nie überschritten!

1 Allgemeines & Überblick

Die Graphenstruktur wird insgesamt in m größere Spalten unterteilt, wobei m als steuerbarer Parameter implementiert werden kann. Zusätzlich werden sog. Synchronisationsgrenzen implementiert, die angeben, nach wie vielen Iterationsschritten die Akkumulatoren an den Spaltenübergängen synchronisiert werden. Dieser Parameter soll ebenfalls von außen steuerbar sein.

Klassenstruktur: Ein Graph wird dynamisch aufgebaut. Dafür gibt es eine Klasse *klasse*. Ein Knoten entsteht, sobald sein Wert positiv wird. Die Knoten werden in einer Oberklasse *Oberklasse* zusammengesetzt. Zur spaltenweise Abarbeitung wird es eine Klasse *Column*? geben. Jede Column bekommt nach Erstellung einen Thread zugewiesen, der die Berechnungen in jeder Iteration übernimmt.

2 Nebenläufige Probleme

Hier gehen wir folgenden Weg:

- Die Synchronisation soll anhand eines Parameters erfolgen. Falls diese Bedingung erfüllt ist, werden mittels Shared Memory die Daten der Akkumulatoren ausgetauscht. Zwei oder mehrere Zyklisches Verhalten bei seltenem Propagieren? Wie handhaben ? -> Testen ob Akkumulatorgröße für Synchronisation von Bedeutung ist.
- Wie wird die Synchronisationsbedingung $i = k$ (Iterationsstadien unterschiedlicher Spalten) sichergestellt?
- Globale / Lokale Konvergenz: alle Threads erreichen ein Maximum an Iterationen ? Alle Spalten bekommen eine eigene Approximationsgrenze ?

3 Verwendete Datenstruktur

Einzelne Spalten der Graphstruktur werden anhand eines Parameters zu mehreren "Threadspalten" zusammengefasst. Eine Threadspalte wird von einem einzelnen Thread bearbeitet. Dabei synchronisiert dieser Thread lokal zwischen den Graphspalten nach jeder Iteration. Zwischen mehreren Threadspalten wird nach einer gewissen Zeit synchronisiert, die von einem äußeren Parameter abhängt. Dazu hat jede Threadspalte eine Referenz auf ihre Nachbarn und kann unabhängig von anderen Threadspalten mit ihren Nachbarn synchronisieren. Jede Threadspalte kennt ihre Größe und kann daraus ableiten, ob bei einer Synchronisation ein neuer Knoten eine neue Threadspalte startet, oder in eine bereits existierende eingefügt wird. Ein Knoten ist ein Objekt, das nur seinen aktuellen Wert kennt. Die Übergangsrate ist in der GraphInfo Klasse enthalten.