

Parallelisierbar = geringer Platzverbrauch

Begriffe

- **2-Band-Turingmaschine:** Zustandsbasierte Maschine mit Eingabeband, auf das nur lesend zugegriffen wird, und Arbeitsband, auf das lesend und schreibend zugegriffen werden kann, sowie einem ausgezeichneten Zustand, der als akzeptierend bezeichnet wird.
- **L : Logarithmischer Platz.** Eine Sprache ist in der Klasse L , wenn eine deterministische 2-Band-Turingmaschine existiert, welche eine Eingabe x genau dann akzeptiert, falls x ein Wort dieser Sprache ist, und deren Arbeitsband auf eine Länge $\mathcal{O}(\log n)$ beschränkt ist.
- **NL : Nichtdeterministischer logarithmischer Platz.** Eine Sprache ist in der Klasse NL , wenn eine nicht-deterministische 2-Band-Turingmaschine existiert, welche eine Eingabe x genau dann akzeptiert, falls x ein Wort dieser Sprache ist, und deren Arbeitsband auf eine Länge $\mathcal{O}(\log n)$ beschränkt ist.
- **AC^1 :** Eine Sprache liegt in AC^1 genau dann, wenn es eine logspace-uniforme Schaltkreisfamilie polynomieller Größe mit maximaler Tiefe $\mathcal{O}(\log n)$ gibt, die entscheidet, ob ein gegebenes Wort zu dieser Sprache gehört oder nicht.
- **NC^1 :** Eine Sprache liegt in NC^1 genau dann, wenn es eine logspace-uniforme Schaltkreisfamilie polynomieller Größe mit maximaler Tiefe $\mathcal{O}(\log n)$ aus Gattern mit höchstens zwei Eingängen gibt, die entscheidet, ob ein gegebenes Wort zu dieser Sprache gehört oder nicht.
- Eine **Schaltkreisfamilie** ist eine Menge von Schaltkreisen, die für jede natürliche Zahl l genau einen Schaltkreis mit l Eingängen enthält. Eine Schaltkreisfamilie heißt **logspace-uniform**, wenn es eine Logspace-Maschine gibt, die zu gegebenem l den entsprechenden Schaltkreis (in geeigneter Codierung) ausgibt.

Behauptung

$$NC^1 \subseteq L \subseteq NL \subseteq AC^1$$

Beweis: $NC^1 \subseteq L$

- Idee: Ergebnis des NC^1 -Schaltkreises rekursiv berechnen mit einer Logspace-Maschine (2-Band-Turingmaschine mit logarithmischer Arbeitsbandbeschränkung)
- Da der Schaltkreis als logspace-uniform vorausgesetzt wurde, existiert eine Logspace-Maschine, die den Schaltkreis codiert liefert. Man kann also die einzelnen Gatter maschinell betrachten.
- Beginne beim Ausgabe-Gatter.
- Werte das Gatter entsprechend der maximal zwei Eingangsleitungen und seines Typs aus.
- Sofern die an den Eingängen anliegenden Werte nicht bekannt sind, verfolge sie jeweils zu demjenigen Gatter, aus dem sie kommen, und berechne den Wert dieses Gatters (Rekursionsaufruf); falls es eine Eingangsleitung des gesamten Schaltkreises ist, lies den Wert vom Eingabeband (Rekursionsabbruch).
- Sobald beide Eingangswerte bekannt sind, kann der Ausgangswert des Gatters ermittelt und zurückgegeben werden.
- Während der gesamten Abarbeitung wird nur $\mathcal{O}(\log n)$ Arbeitsspeicher benötigt: Da es sich um einen flachen Schaltkreis handelt, hat dieser die Tiefe $\mathcal{O}(\log n)$, somit gibt es maximal so tiefe Rekursion. In jeder Rekursionsebene wird konstant viel Information gespeichert: Position des aktuell betrachteten Gatters, Wert an Eingangsleitung 1, Wert an Eingangsleitung 2. Sobald ein Gatter-Ausgabewert berechnet ist und als Eingangswert an den darunterliegenden zurückgegeben worden ist, werden die Informationen dieses beendeten Rekursionsschrittes nicht mehr benötigt und können verworfen werden.
- Da Schaltkreise keine Rückkopplungen (d. h. graphentheoretisch keine gerichteten Kreise) enthalten und endlich groß sind, terminiert die angegebene Logspace-Maschine.

Beweis: $L \subseteq NL$

Klar. Jede deterministische Turingmaschine ist insbesondere nichtdeterministisch.

Beweis: $NL \subseteq AC^1$

- Idee: Kanten im Konfigurationsgraphen einer (nichtdeterministischen) Logspace-Maschine durch Gatter modellieren und Erreichbarkeit in diesem Graphen durch nachgeschaltete Gatter ermitteln.
- Die Konfiguration einer 2-Band-Turingmaschine ist das Tupel bestehend aus der Eingabe (inkl. Position des Lesekopfes auf dem Eingabeband), ihrem inneren Zustand zu einem bestimmten Zeitpunkt und dem aktuellen Zustand des Arbeitsbandes, inkl. der aktuellen Position des Lese-/Schreibkopfes.
- Der Konfigurationsgraph hat als Knoten alle erdenklichen Konfigurationen, und eine (gerichtete) Kante existiert zwischen zwei Knoten (Konfigurationen) genau dann, wenn die Maschine mittels ihrer Übergangsrelation von der einen zur anderen Konfiguration übergehen kann.
- Betrachte nun Konfigurationsgraphen zu beliebiger, aber fester Eingabe. D. h. insbesondere existieren nur Konfigurationsknoten mit derselben Eingabebelegung, sodass diese aus den Tupeln weggelassen werden kann (bis auf die Position des Lesekopfes). Dieser Konfigurationsgraph hat polynomielle Größe (Knotenzahl).
- Für jedes (geordnete) Paar von Knoten setze ein Gatter, das angibt, ob eine Kante vom einen zum anderen Knoten existiert (Ausgangswert 1) oder nicht (Ausgangswert 0). Manche Kanten können unabhängig von der Eingabe immer existieren oder nie existieren – in diesem Fall braucht das Gatter keine Eingangsleitungen. Falls die Existenz einer Kante von dem Eingabebit an der Position des Lesekopfes abhängig ist, so hat das Gatter als Eingangsleitung das betreffende Eingabebit und gibt entweder das Eingangssignal oder seine Negation aus, je nachdem, ob die entsprechende Kante bei Eingabebit 1 oder bei Eingabebit 0 existieren würde.
- Durch Nachschalten weiterer Gatter kann nun aus dieser Adjazenzinformation eine allgemeine Erreichbarkeitsinformation gebildet werden, sodass am Ende für jedes geordnete Paar von Knoten ein Gatter existiert, dessen Ausgabe angibt, ob überhaupt ein Pfad vom einen zum anderen Knoten im Konfigurationsgraphen existiert. Dies wird durch Hintereinanderschalten logarithmisch vieler Pfad-Halbierungsschaltungen realisiert, von der die i -te Schicht wie folgt arbeitet:
 - Eingänge: Für jedes geordnete Paar von Knoten (a, b) einer, der angibt, ob es einen Pfad der Länge $\leq 2^{i-1}$ von a nach b gibt.

- Ausgänge: Für jedes geordnete Paar von Knoten (a, b) einer, der angibt, ob es einen Pfad der Länge $\leq 2^i$ von a nach b gibt.
 - Die Schaltung bestimmt für jedes Paar von Knoten (a, b) , ob sie bereits in der Eingabe einen Pfad besitzen oder ob sie über einen dritten Knoten c zu einem Pfad verbunden werden können, indem für alle Knoten c geprüft wird, ob es in der Eingabe bereits einen Pfad von a nach c und einen Pfad von c nach b gibt. Dann wäre die Verkettung dieser beiden Pfade der Länge $\leq 2^{i-1}$ ein neuer Pfad der Länge $\leq 2 * 2^{i-1} = 2^i$.
 - Realisiert wird diese Schaltung bei Gesamtknotenzahl m also pro geordnetem Paar von Knoten durch $m - 2$ UND-Verknüpfungen mit je zwei Eingängen (für jeden Knoten verschieden von a und b eine, die prüft, ob über ihn ein Pfad zwischen a und b existiert) und einer ODER-Verknüpfung mit $m - 1$ Eingängen, die prüft, ob eine der UND-Verknüpfungen positive Ausgabe hat oder in der Eingabe für die Knoten a und b bereits ein vorhandener Pfad angegeben ist. Wegen $m - 1$ Eingängen an der ODER-Verknüpfung ist dies kein NC^1 -Schaltkreis, sondern lediglich ein AC^1 -Schaltkreis.
- Sei n die Eingabelänge. Dann existieren $\mathcal{O}(n^k)$ Knoten im Konfigurationsgraphen für ein $k \in \mathbb{N}$. Damit gibt es $\mathcal{O}((n^k)^2) = \mathcal{O}(n^{2k})$ geordnete Paare von Knoten und somit wiederum $\mathcal{O}(n^{2k}(1+(n^{2k}) * \lceil \log_2(n^{2k}) \rceil)) \subseteq \mathcal{O}(n^{4k+1})$ Gatter im Schaltkreis – er hat also polynomielle Größe.
 - In einem Graphen mit m Knoten gibt es für zwei Knoten, die über einen Pfad miteinander verbunden sind, stets einen Pfad der Länge $\leq m - 1$, der die beiden Knoten verbindet (indem man einfache Kreise nicht beschreitet). Nach Hinzufügen der i -ten Schicht (wie oben beschrieben) sind alle Pfade der Länge $\leq 2^i$ gefunden. Es werden also höchstens $\lceil \log_2(m - 1) \rceil$ zusätzliche Schichten benötigt. Insgesamt ergeben sich $\lceil \log_2(m - 1) \rceil + 1$ Schichten – die Tiefe des Schaltkreises ist mit $\mathcal{O}(\log m) = \mathcal{O}(\log(n^k)) = \mathcal{O}(k \log n) = \mathcal{O}(\log n)$ logarithmisch, wie benötigt.
 - An der Ausgangsleitung der letzten Schicht, die angibt, ob ein Pfad von der Startkonfiguration zur akzeptierenden Endkonfiguration (die beide o. B. d. A. als eindeutig angenommen werden können) existiert, ist nun abzulesen, ob die nichtdeterministische Logspace-Maschine die anliegende Eingabe akzeptieren würde oder nicht.