DOUBLE PUSHOUT APPROACH

Lanlan Zhang Shu Yang

17. September 2004

Inhaltsverzeichnis

1 Grundbegriffe der direkten Ableitungen

1.1 Graphen, Produktionen und Ableitungen

Zuerst sprechen wir über die Grundbegriffe des algebraischen Ansatzes. Man wendet eine Produktion $p : L \rightarrow R$ als eine endliche schematische Beschreibung von möglichen unendlichen direkten Ableitungen an. L und R sind die linke Seite und die rechte Seite in der Produktion p. Wenn ein Matching m ein Vorkommen von L in einem gegebenen Graph G festlegt, dann bedeutet $G \overset{p,m}{\Longrightarrow} H$ eine direkte Ableitung, dabei führt p Graph G zu einem abgeleiteten Graphen H. Man bekommt den Graph H durch die Ersetzung des Elementes von L in G durch R. Was ist ein Graph? Wie kann man L auf einen Untergraph von G abbilden? Wie kann man die Ersetzung von L durch R in G definieren? Mit diesen Grundbegriffen fangen wir an.

Ein Graph besteht aus zwei Teilen: Träger und Operationen. Knoten **V** und Kanten **E** sind die Träger. $s : E \rightarrow V$ und $t : V \rightarrow E$ sind zwei einstellige Operationen. Es gibt noch markierte Funktionen $L_v : V \rightarrow L_v$und $L_e : E \rightarrow L_e$, wobei L_v und L_e feste Kennzeichensalphabeten für Knoten und Kanten sind.

Jede graphische Produktion, die eine teilweise Gleichheit zwischen Elementen von seinen linken und rechten Seite definiert, entscheidet, welche Knoten und Kanten noch behalten werden, welche Knoten und Kanten gelöscht werden und welche Knoten und Kanten erzeugt werden sollen. Um zu sehen, wie es funktioniert, haben wir hier ein Beispiel im **Bild 1.1**. Die Produktion $p_1 : L_1 \rightarrow R_1$ wird auf der linken Seite von **Bild 1.1** auf einen Graphen G_1 angewendet, führt zu einer direkten Ableitung (1). Die Produktion hat zuerst drei Knoten($Bezeichnungen\, 1, 2, 3$) in der linken Seite, nach der Produktion bleibt nur noch zwei davon ($Bezeichnungen\, 1, 3$), eine neue Kante ($Bezeichnungen\, 5$) zwischen Knoten 1 und Knoten 3 wird in der rechte Seite erzeugt. Um die Produktion p_1 in Graph G_1 anzuwenden, müssen wir ein Vorkommen in der linken Seite L_1 in G_1 finden, das nennen wir *Matching*. Ein Matching $m : L \rightarrow G$ für eine Produktion p ist ein Graph-Homomorphismus. Er bildet Knoten und Kanten von L zu G ab. Das Matching $m_1 : L_1 \rightarrow G_1$ in unserem Beispiel bildet das Element von L_1 zu einem Element in Graph G_1 mit dem gleichen Namen ab. Wenden wir die Produktion p_1 zu Graph G_1 auf dem Matching m_1 an, dann müssen wir die Elemente in Graph G_1, die Bilder von L_1 durch m_1 sind und aber keine entsprechende Elemente in R_1 existieren, löschen. Entsprechend müssen wir auch die Elemente, die zu R_1 aber nicht zu L_1 gehören, einfügen. Alle Reste von G_1 bleiben noch, dann haben wir der angeleiteten Graph H_1. Grob gesagt, H_1 ist als $G_1 - (L_1 - R_1) \bigcup (R_1 - L_1)$ konstruiert.

Die Anwendung von einer Produktion kann als eine Einbettung in einem Kontext interpretiert werden. Der Kontext ist ein Teil von Graph G, der aber kein Teil von dem Matching ist. In unserem Beispiel ist er der Kante 4. Auf dem Boden der direkten Ableitung verbindet die co-Produktion $p_1^* : G_1 \rightarrow H_1$ den gegebenen Graph G_1 und abgeleiteten Graph H_1. Eine direkte Ableitung von G zu H wird von der Anwendung der Produktion p mit dem Matching m schematisch als Bild 1.2 dargestellt. Wir bezeichnen das als $d = (G \overset{p,m}{\Longrightarrow} H)$.

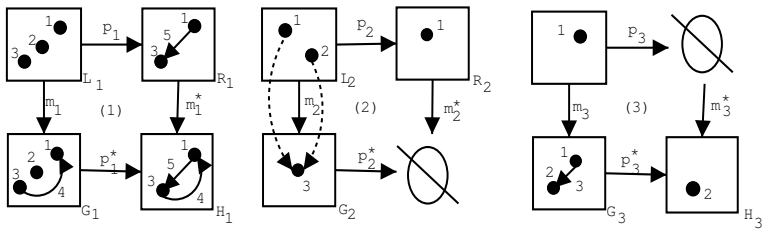


Abbildung 1.1

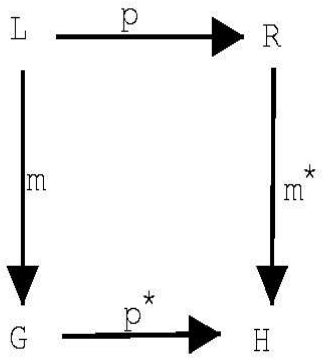


Abbildung 1.2

Es gibt aber Probleme. Beispielweise betrachten wir die direkte Ableitung (2) von Bild 1.1. Die Produktion p_1 bekommt zwei Knoten und löscht einen davon. p_1 wird auf Graph G_2, der nur einen Knoten hat, angewendet. Knoten 1 und 2 von L_2 werden beide zu Knoten 3 in G_2 abgebildet. Aber die Produktion p_2 muss gleichzeitig Knoten 3 löschen und behalten. Wir sagen, dass das Matching m_2 einen Konflikt enthält. Es gibt drei Möglichkeiten, um diese Konflikte zu lösen. Knoten 3 kann behalten werden. Knoten 3 kann gelöscht werden. Und die Produktion p_2 unter dem Matching kann verboten werden. In der direkten Ableitung (2) in Bild 3.1 haben wir die zweite Möglichkeit genommen. Der Knoten 3 wird gelöscht.

Ein Knoten ist mit einer Kante verbunden, die kein Teil des Matchinges ist. Wenn der Knoten gelöscht wird, bekommen wir dann einen Konflikt. In der direkten Ableitung (3) in Bild 3.1 haben wir dieses Problem gezeigt. Wenn wir den Knoten 1 löschen, dann fehlt es bei der Kante 3 ein Knoten, der Rest ist kein Graph mehr. Solche Kanten nennen wir *Hängende Kanten*. Hier haben wir auch zwei Möglichkeiten, um diese Konflikte zu lösen. Wir können die Kante 3 mit Knoten 1 zusammen löschen. Oder wir können auch die Produktion p_3 verbieten. In unserem Beispiel haben wir die Kante 3 auch gelöscht.

In algebraischen Ansätze werden direkte Ableitungen von klebende Konstruktionen der Graphen beschrieben. Die nennen wir Pushout. In Pushout werden die Graphen als Objekt angewendet. Graph-Homomorphismen werden als Pfeil gekennzeichnet. Eine Produktion in DPO wird als ein Paar $L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R$ Graph-Homomorphisms von einem *interface* Graph K gegeben. In DPO besteht eine direkte Ableitung aus zwei klebenden Diagrammen von Graphen und totalen Graph-Homomorphisms. Wir

haben das in Bild 1.3 gezeigt.

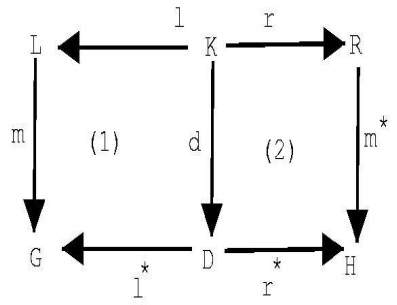


Abbildung 1.3

Man bekommt den Kontext Graph D von dem gegebenen Graph G durch Löschen aller Elmente in G, die ein Urbild in L aber kein Urbild in K hat. Diagramm (2) modelliert die Einfügung in H von allen Elementen, die keine Urbild in K haben. Um die Probleme in (2) und (3) in Bild 1.1 zu vermeiden, muss das Matching m in die hängende Kanten Bedingung erfüllen. Das nennen wir *Pushout Bedingungen*. Die Bedingungen bestehen aus zwei Teilen. Um zu versichern, dass es keine *Hängende Kanten* gibt, verlangt die *Hängende Bedingung*: wenn p einen Knoten löscht, muss sie auch alle Kanten in G, die mit dem Knoten verbunden sind, löschen. Die zweite Art ist die *Identifizierungsbedingung*. Diese Bedingung verlangt, dass jedes Element in G, das durch die Anwendung von p gelöscht wird, nur ein Urbild in L hat.

Eine *Graphgrammatik* $\mathcal{G}$ besteht aus einer Reihe von Produktionen P und einem Startgraph G_0. Eine Sequenz von direkten Ableitungen $\rho = (G_0 \overset{p_1}{\Longrightarrow} G_1 \overset{p_2}{\Longrightarrow} \ldots) \overset{p_n}{\Longrightarrow} G_n$ konstruiert eine Ableitung der Grammatik, geschrieben als $G_o \Longrightarrow^* G_n$. Die Sprache $\mathcal{L}(\mathcal{G})$, die von $\mathcal{G}$ erzeugt, besteht aus allen Graphen G_n, für $G_o \Longrightarrow^* G_n$ eine Ableitung der Grammatik ist.

1.2 Parallele und Sequenzielle Unabhängigkeit

Interleaving Wir werden jetzt den Interleaving-Ansatz betrachten. Wir stellen die Frage in zwei Fällen, ob zwei direkte Ableitungen Konkurrieren.

1. Angenommen, dass ein gegebener Graph einen bestimmten Zustand darstellt. Der nächste Zustand wird durch die Anwendung einer Produktion auf einem Matching erreicht. Dabei werden die Produktion und das Matching nicht deterministisch aus mehreren Alternativen ausgewählt werden. Jede Wahl führt zu einer unterschiedlichen Sequenz. Die Frage ist nur, ob zwei Sequenzen davon tatsächlich unterschiedliche Berechnungen modllieren oder wir nur ein von beiden gleichwertigen Interleaving gewählt haben. Mit anderen Wötern, zwei alternativen direkten Ableitungen $H_1 \overset{p_1}{\Longleftarrow} G \overset{p_2}{\Longrightarrow} H_2$

werden gegeben, die Frage ist, ob es direkten Ableitungen $H_1 \overset{p_2}{\Longrightarrow} G' \overset{p_1}{\Longleftarrow} H_2$ gibt, jede direkte Ableitung davon kann nach Anwendung der Andere verschoben werden, aber wir können den gleichen Resultat bekommen.

2. Es gibt zwei sequenzielle direkten Ableitungen $G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$. Können wir eine andere Sequenz $G \overset{p_2}{\Longrightarrow} H_2 \overset{p_1}{\Longrightarrow} X$ nehmen, und das Resultat gleich bleibt?

Problem 2.1(Local Church Rosser)

1. Finden Sie eine Bedingung, die parallele Unabhängigkeit heißt. Zwei alternativen direkten Ableitungen $H_1 \overset{p_1}{\Longleftarrow} G \overset{p_2}{\Longrightarrow} H_2$ sind parallele unabhängig, wenn es direkte Ableitungen $H_1 \overset{p_2}{\Longrightarrow} X$ und $H_2 \overset{p_1}{\Longrightarrow} X$ gibt, dass $G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$ und $G \overset{p_2}{\Longrightarrow} H_2 \overset{p_1}{\Longrightarrow} X$ equivalent sind.

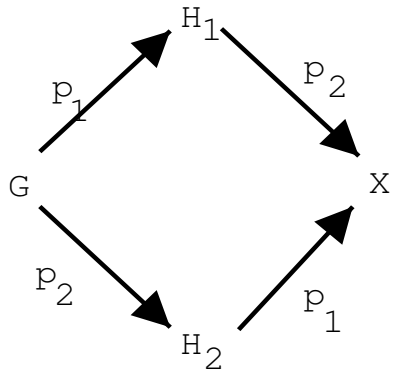


Abbildung 2.1

2. Finden Sie eine Bedingung, die sequenzielle Unabhängigkeit heißt. Eine direkte Ableitung $G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$ ist sequenzielle unabhängig, wenn es eine gleichwertige direkte Ableitung $G \overset{p_2}{\Longrightarrow} H_2 \overset{p_1}{\Longrightarrow} X$ gibt.

Parallelismus

Eine parallele Anwendung kann durch die Anwendung von einzelnen Produktionen modelliert werden, das nennen wir Parallel-Produktion.

Zwei Produktionen $p_1 : L_1 \rightarrow R_1$ und $p_2 : L_2 \rightarrow R_2$ seien gegeben. Wir bezeichnen ihre Zusammensetzung als $p_1 + p_2 : L_1 + L_2 \rightarrow R_1 + R_2$. $p_1 + p_2$ ist eine disjunktive Vereinigung von p_1 und p_2. Eine direkte Ableitung $G \overset{p_1+p_2}{\Longrightarrow} X$ mit der Anwendung einer parallelen Produktion $p_1 + p_2$, nennen wir eine parallele direkte Ableitung.

Jetzt haben wir schon zwei verschiedene Methoden gesehen, die Parallelberechnungen durch Graph-Transformationen modelliert. Die eine ist Interleaving-Ableitung, und die andere ist die parallele direkte Ableitung. Welchen Zusammenhang gibt es

zwischen den beiden? Können wir für eine bestimmte parallele direkte Ableitung $G \overset{p_1+p_2}{\Longrightarrow} X$ eine entsprechende Interleaving-Ableitung $G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$ finden? Können wir für eine bestimmte Interleaving-Ableitung eine parallele direkte Ableitung finden?

Problem 2.2(Parallelismus)

1. Finden Sie eine Sequentialisierungsbedingung. Eine parallel direkte Ableitung $G \overset{p_1+p_2}{\Longrightarrow} X$ befriedigt die Bedingung, wenn es eine äquivalente sequenzielle Ableitung $G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$ (oder $G \overset{p_2}{\Longrightarrow} H_2 \overset{p_1}{\Longrightarrow} X$) gibt.

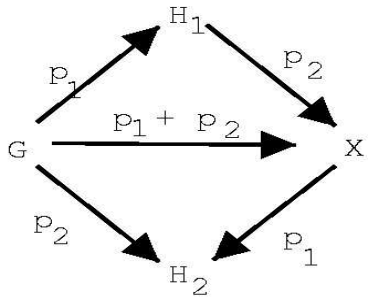


Abbildung 2.2

2. Finden Sie eine Parallelisierungsbedingung. Eine sequenzielle Ableitung $G \overset{p_1}{\Longrightarrow} H_1 \overset{p_2}{\Longrightarrow} X$ befriedigt die Bedingung, wenn es eine equivalente parallele Ableitung $G \overset{p_1+p_2}{\Longrightarrow} X$ gibt.

1.3 Einbettung der Ableitungen und abgeleitete Produktionen

Die Ableitung $\rho = (G_0 \overset{p_1}{\Longrightarrow} \ldots \overset{p_n}{\Longrightarrow} G_n)$ wird zu $\sigma = (X_0 \overset{p_1}{\Longrightarrow} \ldots \overset{p_n}{\Longrightarrow} X_n)$ eingebettet, wenn beide die gleiche Länge haben. Jede Graph G_i von ρ wird in den entsprechenden Graph X_i von σ eingebettet. Die Einbetung wird von Injektionen $(G_i \overset{e_i}{\rightarrow} X_i)_{i\in\{0,\ldots n\}}$ beschrieben, die als $\rho \overset{e}{\rightarrow} \sigma$ bezeichnet werden. Wenn diese Einbettung existiert, dann wird sie eindeutig durch die erste Injektion $e_0 : G_0 \rightarrow X_0$ bestimmt. Jetzt werden ρ und e_0 eingegeben. Die Frage ist nun, unter welche Bedingungen es eine Ableitung σ gibt, so macht e_0 eine Einbettung e von ρ zu σ.

Problem 2.3 (Einbettung)

Gibt es eine Bedingung? Für eine gegebene Ableitung $\rho = (G_0 \stackrel{p_1}{\Longrightarrow} \ldots \stackrel{p_n}{\Longrightarrow} G_n)$ befriedigt eine Injektion $e_0 : G_0 \rightarrow X_0$ diese Bedingung, wenn es eine Ableitung $\sigma = (X_0 \stackrel{p_1}{\Longrightarrow} \ldots \stackrel{p_n}{\Longrightarrow} X_n)$ und eine Einbettung $e : \rho \rightarrow \sigma$ gibt.

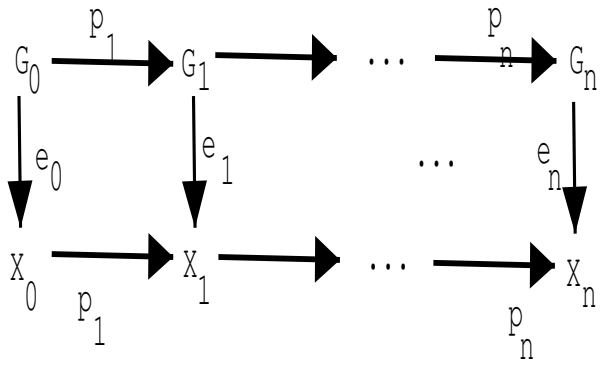


Abbildung 2.3

In Bild 3.2 haben wir eine co-Produktion p^* definiert. Es gibt eine Ableitung $\rho = (G_0 \stackrel{p_1}{\Longrightarrow} \ldots \stackrel{p_n}{\Longrightarrow} G_n)$. Wir haben $p^* : G_0 \rightarrow Gn$ als direkte abgeleitete Produktion definiert.

Problem 2.4(Abgeleitete Produktionen)

Sei $\rho = (G_0 \stackrel{p_1}{\Longrightarrow} \ldots \stackrel{p_n}{\Longrightarrow} G_n)$ eine Ableitung. Finden Sie eine abgeleitete Produktion $p^* : G_0 \rightarrow Gn$. Für jede Injektion $G_0 \stackrel{e_0}{\rightarrow} X_0$ gibt es eine direkte Ableitung $X_0 \stackrel{p^*}{\Longrightarrow} X_n$ auf e_0, wenn e_0 eine Einbettung $e : \rho \rightarrow \sigma$ von ρ zu einer Ableitung $\sigma = (X_0 \stackrel{p_1}{\Longrightarrow} \ldots \stackrel{p_n}{\Longrightarrow} X_n)$ macht.

2 Graph Transformation basierend auf der DPO Konstruktion

Definition 3.1(Markierte Graphen)
Ω_V und Ω_E sind Alphabete für Knoten und Kanten. Ein markierter Graph bezeichnen wir als $G = \langle G_V, G_E, s^G, t^G, l_v^G, l_e^G \rangle$. G_V und G_E sind die Menge für Knoten und Kanten.$s^G, t^G : G_E \rightarrow G_V$ sind die Ursprungs- und Zielfunktion. $l_v^G : G_V \rightarrow \Omega_V$ und $l_e^G : l_e^G \rightarrow \Omega_E$ sind die markierten Funktionen für Knoten und Kanten.
Ein Graph-Homomorphisms $f : G \rightarrow G'$ ist ein 2-Tuple $f : \langle f_V : G_V \rightarrow G'_V, f_E : G_E \rightarrow G'_E \rangle$ von Funktionen, die $f_V \circ t^G = t^{G'} \circ f_E, f_V \circ s^G = s^{G'} \circ f_E, l_v^{G'} \circ f_V = l_v^G$ und $l_e^{G'} \circ f_E = l_e^G$ befriedigen. Ein Graph-Homomorphisms ist ein Isomorphismus , wenn f_V und f_E Bijektion sind. Wenn ein Isomorphismus von Graph G zu Graph H existiert, dann schreiben wir $G \cong H$. Ferner, $[G]$ bedeutet die Isomorphismusklasse von G,

$[G] = \{H | H \cong G\}$. Ein Automorphismus von Graph G ist ein Isomorphismus $\phi : G \rightarrow G$. Die Kategorie mit markierten Graphen als Objekte und Graphmorphismen als Pfeile nennen wir **Graph**.

Beispiel 3.2(Client-Server System)
In folgenden Beispiel stellen die mit C markierten Knoten Client und die mit S markierten Knoten Server dar. Ein mit *idle* markierter Kreis auf einem Server zeigt, dass der Server breit ist, um eine Abfrage zu akzeptieren. Ein mit *job* markierter Kreis auf einem Client zeigt, dass der Client zurzeit arbeitet. Ein mit *req* markierter Kreis auf einem Client zeigt, dass der Client eine Abfrage gesendet hat. Eine mit *busy* markierte Kante von einem Client zu einem Server zeigt, dass der Server die Abfrage vom Client verarbeitet. In unserem Beispiel sind $\Omega_V = \{C, S\}$ und $\Omega_E = \{idle, job, req, busy\}$ Arphabete. Bild 3.1(a) zeigt einen Graph $G = \langle G_V, G_E, s^G, t^G, l_v^G, l_e^G \rangle$, und $G_V = \{0, 1, 2, 3, 4\}, G_E = \{\underline{0}, \underline{1}, \underline{2}, \underline{3}, \underline{4}\}$. Bild 3.1($b$) zeigt die Isomorphismus-Klasse von G.

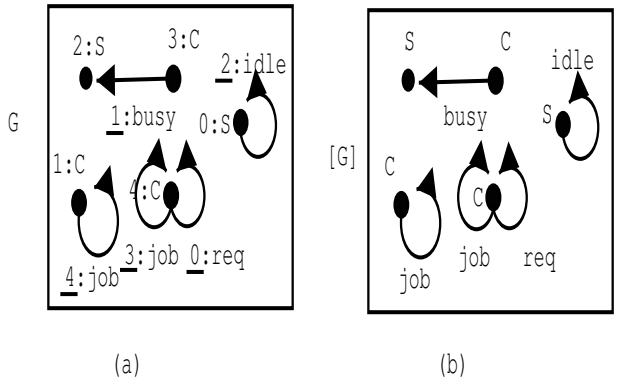


Abbildung 3.1

Definition 3.3(Graph Produktionen, Graph Grammatik)
Eine Graph-Produktion $p : (L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R)$ besteht aus einem *Produktionsnamen* p und zwei injektiven Graph-Homomorphisms $l : K \rightarrow L$ und $r : K \rightarrow R$. Die Graphen L,K und R nennen wir die *linke Seite*, *die Interface* und *die rechte Seite* von p. Zwei Produktionen $p : (L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R)$ und $p' : (L' \overset{l'}{\leftarrow} K' \overset{r'}{\rightarrow} R')$ sind Span-Isomorphisms, wenn es drei Isomorphismen $L \overset{\phi_L}{\rightarrow} L', K \overset{\phi_K}{\rightarrow} K'$ und $R \overset{\phi_R}{\rightarrow} R'$, die zwei Quadrate konstruieren, gibt. Die drei Isomorphismen befriedigen auch $\phi_L \circ l = l' \circ \phi_K$ und $\phi_R \circ r = r' \circ \phi_K$. Wir schreiben $L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R$ anstatt $p : (L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R)$.
Eine Graphgrammatik $\mathcal{G}$ ist ein paar $\mathcal{G} = \langle (p : L \overset{l}{\leftarrow} K \overset{r}{\rightarrow} R)_{p \in P}, G_0 \rangle$. Die ersten Komponenten sind eine Familie von Produktionen mit den Namen in P. G_0 ist der *Startgraph*.

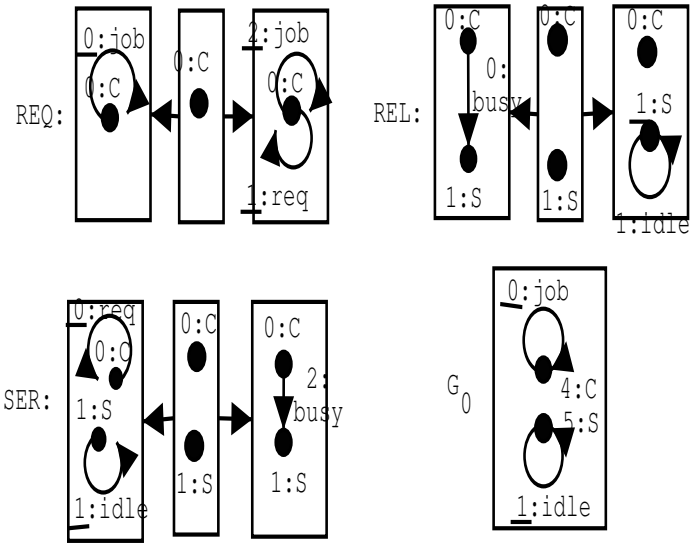


Abbildung 3.2

Beispiel 3.4 (Graph Grammatik für Client-Server Systeme) In unserem Beispiel gibt es drei Produktionen: **REQ**, **SER** und **REL**. Produktion REQ beschreibt die Ausgabe einer Abfrage von einem Client. Nachdem die Abfrage erzeugt wird, dann arbeitet der Client weiter, und die Abfrage wird asynchron verarbeitet. In Produktion SER verbindet ein Client, der eine Abfrage gesendet hat, durch eine *busy* Markierte Kante mit einem Server, der auf einer Abfrage wartet. Produktion REL trennt einen Client von einem Server, danach bleibt der Server in *idle* Zustand. Die Grammatik nennen wir **C-S**, und es wird als **C-S** $= \langle \{REQ, SER, REL\}, G_0 \rangle$ definiert. G_0 ist der Startgraph.

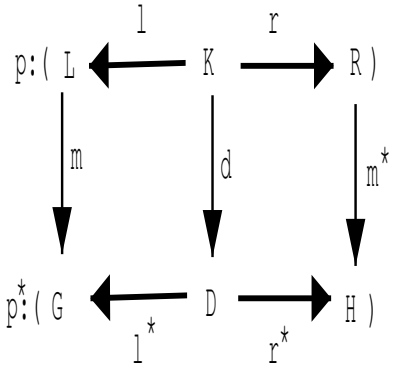


Abbildung 3.3

Definition 3.4 (Pushout und Pushout Komplement)
Eine Kategorie **C** und zwei Pfeile $b : A \rightarrow B, c : A \rightarrow C$ von **C** seien gegeben. Eine

3-Tupel $\langle D, g : B \rightarrow D, f : C \rightarrow D \rangle$ wie das Diagramm unten nennen wir ein *Pushout* von $\langle b, c \rangle$, wenn

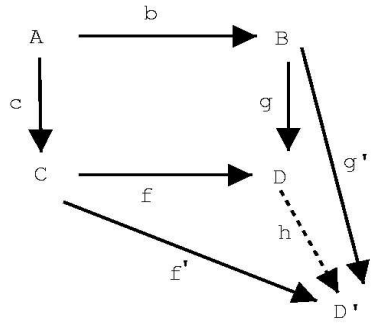


Abbildung 3.4

Kommutivität: $g \circ b = f \circ c$ und
Universale Eigenschaft: Für alle Objekte D' und Pfeile $g' : B \rightarrow D'$ und $f' = C \rightarrow D'$, mit $g' \circ b = f' \circ c$. Es existiert nur eine Pfeil $h : D \rightarrow D'$, so $h \circ g = g'$ und $h \circ f = f'$.

D nennen wir ein *Pushout-Objekt* von $\langle b, c \rangle$. Ferner, $b : A \rightarrow B$ und $g : B \rightarrow D$ seien gegeben, eine 3-Tupel $\langle C, c : A \rightarrow C, f : C \rightarrow D \rangle$ ist *ein Pushout Komplement* von $\langle b, g \rangle$. **C** nenne wir ein *Pushout Komplement Objekt* von $\langle b, g \rangle$.

Definition 3.5 (Direkte Ableitung)
Einer Graph G, eine Graph Produktion $p : (L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R)$ und ein *Matching* $m : L \rightarrow G$ seien gegeben. Eine direkte Ableitung von G zu H durch p auf m existiert, wenn ein Diagramm wie Bild 3.5 konstruiert werden kann. Dabei ist es erforderlich, dass beide Quadrate Pushout in Graph sind. Wir nennen D Kontextgraph. Wir schreiben $G \stackrel{p,m}{\Longrightarrow} H$ oder $G \stackrel{p}{\Longrightarrow} H$.

Beispiel 3.6
Bild 3.5 zeigt eine direkte Ableitung $G_0 \stackrel{REQ}{\Longrightarrow} G_1$.

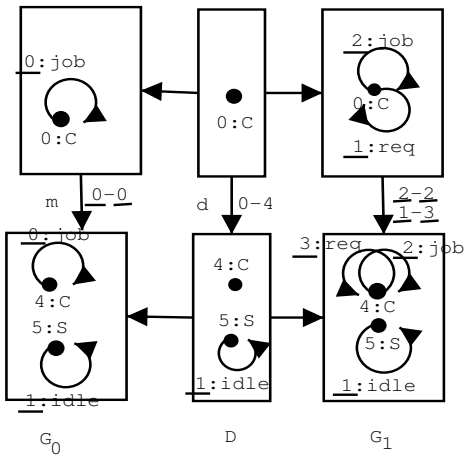


Abbildung 3.5

Definition 3.7 (Sequenzielle Ableitung)
Eine *sequenzielle Ableitung* ist entweder ein Graph G, dann nennen wir eine *Identität Ableitung* und schreiben als $G : G \Longrightarrow^* G$, oder eine Sequenz von direkten Ableitungen $\rho = \{G_{i-1} \overset{p_i}{\Longrightarrow} Gi\}_{i\in\{1,\ldots n\}}$, p_i ist eine Produktion von $\mathcal{G}$ für alle $i \in \{1, \ldots n\}$.Im zweiten Fall schreiben wir die Ableitung als $\rho : G_0 \Longrightarrow^*_{\mathcal{G}} G_n$ oder einfach $\rho : G_0 \Longrightarrow^* G_n$. Wenn $\rho : G \Longrightarrow^* H$ eine Ableitung ist, dann sind Graphen G und H der *Start-* und der *Endgraph* von ρ, und werden als $\sigma(\rho)$ und $\tau(\rho)$ geschrieben. Die *Länge* der sequenziellen Ableitung ρ ist die Anzahl der direkten Ableitungen in ρ. Die *sequenzielle Zusammensetzung* von zwei Ableitungen ρ und ρ' werden nur definiert, wenn $\tau(\rho) = \sigma(\rho')$, dann schreiben wir es als $\rho; \rho' : \sigma(\rho) \Longrightarrow^* \tau(\rho')$.

3 Unabhängigkeit und Parallelität für den DPO Ansatz

3.1 Parallele und Sequenzielle Unabhängigkeit

In diesem Abschnitt werden zwei Notationen für Unabhängigkeit der direkte Ableitung definiert. Zuerst betrachten wir die *parallele Unabhängigkeit*. Zwei alternative direkte Ableitungen $H_1 \overset{p_1,m_1}{\Longleftarrow} G \overset{p_2,m_2}{\Longrightarrow} H_2$ sind *parallel unabhängig*, falls jede nach der Ausführung einer anderen doch angewandet werden kann. Das heißt, weder die Ableitung $G \overset{p_1,m_1}{\Longrightarrow} H_1$ noch noch die $G \overset{p_2,m_2}{\Longrightarrow} H_2$ kann Elemente von G löschen, die von der anderen Ableitung benötigt werden. Mit anderen Wörten, die Überscheidung der linken Seiten von p_1 und p_2 in G, also $m_1(L_1)$ und $m_2(L_2)$, muss im Durchschnitt der entsprechenden Interfacegraphen K_1 und K_2 enthalten sein.

Definition 4.1(Parallele Unabhängigkeit)
Seien $G \overset{p_1,m_1}{\Rightarrow} H_1$ und $G \overset{p_2,m_2}{\Rightarrow} H_2$ zwei direkte Ableitungen wie in der Abbildung 3.8.

Sie sind *parallel unabhängig*, falls gilt

$$m_1(L_1) \cap m_2(L_2) \subseteq m_1(l_1(K_1)) \cap m_2(l_2(K_2)).$$

Diese Eigenschaft kann auch so formuliert werden:
Es existieren zwei Graphmorphismen $L_1 \xrightarrow{k_2} D_2$ und $L_2 \xrightarrow{k_1} D_1$ mit

$$l_2^* \circ k_2 = m_1 \text{ und } l_1^* \circ k_1 = m_2.$$

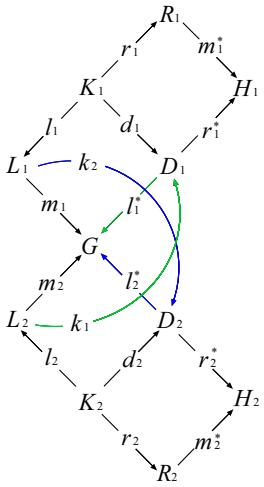


Abbildung 4.1: Parallele Unabhängigkeit der direkte Ableitung

Zwei aufeinander folgende direkte Ableitungen $G \overset{p_1,m_1}{\Longrightarrow} H_1 \overset{p_2,m_2}{\Longrightarrow} H_2$ sind *sequentiell unabhängig*, wenn sie vertauschbar sind. Das heißt, p_2 auf der Matching m_2 darf keine von p_1 auf der Matching m_1 unveränderte Elemente löschen und keine von p_1 auf der Matching m_1 erzeugte Elemente benutzen. Mit anderen Wörten, die Überscheidung von R_1 und L_2 in H_1 muss im Durchschnitt der entsprechenden Interfacegraphen K_1 und K_2 enthalten sein.

Definition 4.2 (Sequentielle Unabhängigkeit)
Eine Transformationssequenz $G \overset{p_1,m_1}{\Longrightarrow} H_1 \overset{p_2,m_2}{\Longrightarrow} H_2$ wie in der Abbildung 3.9 ist *sequentiell unabhängig*, falls gilt

$$m_1^*(R_1) \cap m_2(L_2) \subseteq m_1^*(r_1(K_1)) \cap m_2(l_2(K_2)).$$

Diese Eigenschaft kann auch so formuliert werden:
Es existieren zwei Graphmorphismen $R_1 \xrightarrow{k_2} D_2$ und $L_2 \xrightarrow{k_1} D_1$ mit der Eigenschaft

$$l_2^* \circ k_2 = m_1^* \text{ und } r_1^* \circ k_1 = m_2.$$

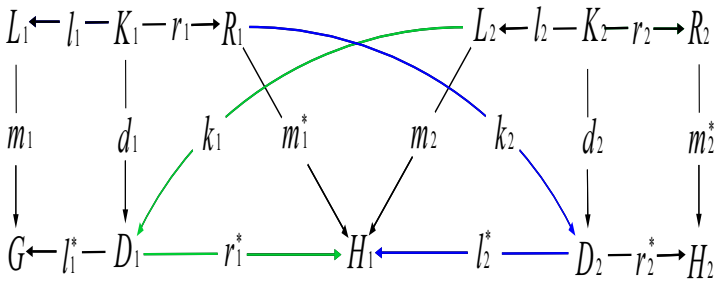


Abbildung 4.2: Sequentielle Unabhängigkeit der direkte Ableitung

Beispiel 4.3 (Sequentielle Unabhängigkeit)
Jetzt betrachten wir die ersten zwei direkten Abbildungen in der Abbildung 3.7. Die zweite Produktion `SER` auf dem zweiten Matching benutzt die Kante $\underline{3} : seq$ im Graph G_1, die nicht im Graph G_0 enthalten ist sondern von der ersten Produktion `REQ` auf dem ersten Matching erzeugt wird. Das bedeutet:

$$\underline{3} \in m_1^*(R_1) \cap m_2(L_2) \text{ aber } \underline{3} \in m_1^*(r_1(K_1)) \cap m_2(l_2(K_2)).$$

Also die zwei direkte Ableitung sind *sequentiell abhängig*. Aus der foldende Abbildung ist es auch klar, dass $G_0 \overset{\texttt{SER}}{\Rightarrow} G_1' \overset{\texttt{REQ}}{\Rightarrow} G_2$, also die zwei Produktionen `REQ` und `SER` sind nicht zu G_0 vertauschbar (*sequentiell abhängig*).

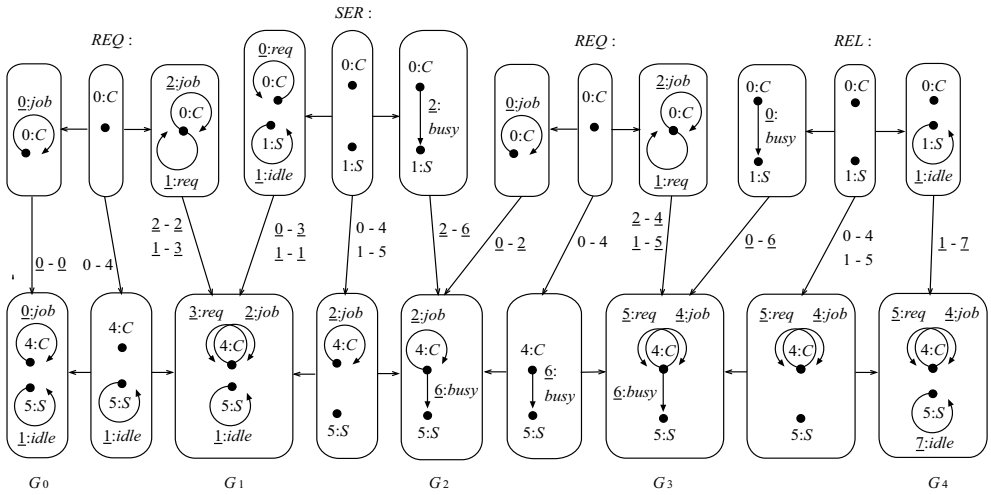


Abbildung 4.3

In der Abbildung 4.3 ist es einfach zu sehen, dass die Ableitung von G_1 zu G_3 und die von G_2 zu G_4 *sequentiell unabhängig* sind.

Definition 4.4 (Koproduktionen)
Seien **C** eine Kategorie und A,B zwei Objekte in **C**. Eine *Koproduktion* von $\langle A, B\rangle$ ist ein 3-Tuple $\langle A+B, in_1^{A+B}, in_2^{A+B}\rangle$ wobei $A+B$ ein Objekt und $in_1^{A+B} : A \to A+B$,

$in_2^{A+B} : B \to A + B$ Pfeile in $\mathbf{C}$ sind, so dass es genau ein Pfeil $f : A + B \to C$ für je zwei Pfeile $\langle f_1 : A \to C, f_2 : B \to C \rangle$ existiert und $f \circ in_1^{A+B} = f_1$, $f \circ in_2^{A+B} = f_2$. In diesem Fall heißt $A + B$ ein *Koproduktionsobjekt* von $\langle A, B \rangle$.

Definition 4.5 (Parallele Produktionen)
Sei $\mathcal{G}$ eine Graphgrammatik. Eine parallele Produktion ist gegeben durch

$$\langle (p_1, in^1), \ldots, (p_k, in^k) \rangle : (L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R)$$

(siehe Abbildung 4.4), wobei
- $k \geq 0$,
- $p_i : (L_i \stackrel{l_i}{\leftarrow} K_i \stackrel{r_i}{\rightarrow} R_i)$ Produktionen von $\mathcal{G}$ für alle $i \in \{1, \ldots, k\}$,
- L, R bzw. K die Koproduktionsobjekte von $\langle L_1, \ldots, L_k \rangle$, $\langle R_1, \ldots, R_k \rangle$ bzw. $\langle K_1, \ldots, K_k \rangle$,
- l und r eindeutig durch die Familie der Graphmorphismen $\{l_i\}_{i \leq k}$ und $\{r_i\}_{i \leq k}$ gegeben sind.

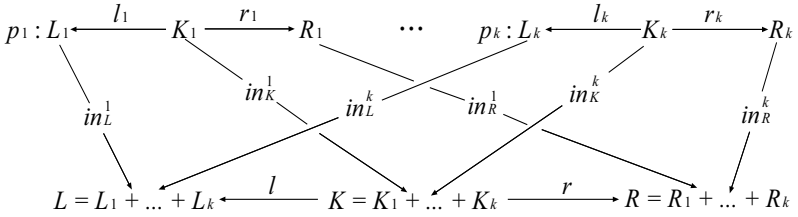


Abbildung 4.4: Die parallele Produktion $\langle (p_1, in^1), \ldots, (p_k, in^k) \rangle : (L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R)$

Die von einer Graphgrammatik $\mathcal{G}$ erzeugte Graphgrammatik mit parallelen Produktionen $\mathcal{G}^+$ hat dieselbe Quellgraph wie $\mathcal{G}$ und alle parallele Produktionen von $\mathcal{G}$ als Produktion. Eine *parallele direkte Ableitung auf $\mathcal{G}$* ist dann eine direkte Ableitung auf $\mathcal{G}^+$.

Beispiel 4.6 (Parallele direkte Ableitung)
Die Abbildung 3.11 zeigt eine parallele direkte Ableitung, die auf der parallelen Produktion `SER + REQ` und einem nicht-injektiven Matching basiert. Diese parallele direkte Ableitung modelliert die Situation, dass der Client 4 eine neue Anfrage sendet während der Server 5 anfängt, eine alte Anfrage zu bearbeiten.

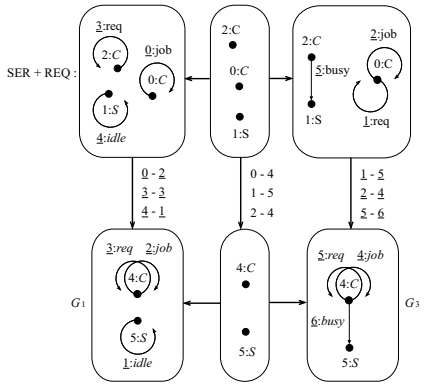


Abbildung 4.5: Eine parallele direkte Ableitung

3.2 Local Church Rosser Problem und Parallelitiät

Satz 4.7 (Anwendung der parallelen Produktionen)
Sei $q = \langle (p_1, in^1), \ldots, (p_k, in^k) \rangle : (L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R)$ eine parallele Produktion, und $L \stackrel{m}{\rightarrow} G$ ein Matching. Sei $L_i \stackrel{m_i}{\rightarrow} G$ Matching der Produktion p_i, wobei $m_i = m \circ in_L^i$, für alle $i \in \{1, \ldots, k\}$, siehe folgende Abbildung:

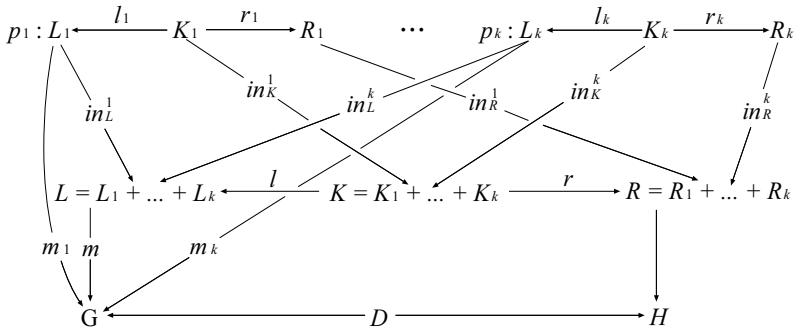


Abbildung 4.6

Dann gibt es eine parallele direkte Ableitung $G \stackrel{q,m}{\Rightarrow} H$, genau dann wenn für alle $1 \leq i < j \leq k$ gilt:
• $\langle l_i, m_i \rangle$ erfüllt die Gluing-Kondition, d.h, es existiet eine direkte Ableitung $G \stackrel{p_i,m_i}{\Rightarrow} H_i$,
• die direkten Ableitungen $G \stackrel{p_i,m_i}{\Rightarrow} H_i$ und $G \stackrel{p_j,m_j}{\Rightarrow} H_j$ sind parallel unabhängig.

Satz 4.7 zeigt den wichtigen Zusammenhang zwischen *paralleler Unabhängigkeit* und *parallelen dirketen Ableitungen*: verschiedene Matchings der Produktionen in einer Graph führt zu einem Matching der entsprechenden parallelen Produktion, genau dann wenn sie paarweise parallel unabhängig sind. Im nächsten Kapitel werden wir sehen, dass dieser Aussage für den SPO Ansatz nicht mehr richtig ist.
Die folgende zwei Lemma sagen aus, dass jede parallele direkte Ableitung in eine Sequenz sequentiell unabhängiger direkter Ableitungen transformiert werden kann, und umgekehrt, dass jede sequentiell unabhängige Ableitung in eine parallele direkte Ableitung transformiert werden kann.

Lemma 4.8 (Analysis der parallelen direkten Ableitungen)
Sei $q = \langle (p_1, in^1), \ldots, (p_k, in^k) \rangle : (L \stackrel{l}{\leftarrow} K \stackrel{r}{\rightarrow} R)$ eine parallele Produktion, und $\rho = (G \stackrel{q}{\Rightarrow} H)$ eine parallele direkte Ableitung.
Dann gibt es eine sequentiell unabhängige Ableitung $\rho' = (G \stackrel{q'}{\Rightarrow} X \stackrel{q''}{\Rightarrow} H)$, heißt eine *Analysis* von ρ, wobei

$$q' = p_{i_1} + \ldots + p_{i_n} \text{ und } q'' = p_{j_1} + \ldots + p_{j_m}$$

für alle Partitionen $\langle I = \langle i_1, \ldots, i_n \rangle, J = \langle j_1, \ldots, j_m \rangle \rangle$ von $\{1, \ldots, k\}$ (also $I \cup J = \{1, \ldots, k\}$ und $I \cap J = \emptyset$). In diesem Fall schreiben wir $\rho' \in ANAL_{I,J}(\rho)$.
Bemerkung: Diese Ausführung ist im Allgemein nicht deteministisch.

Lemma 4.9 (Synthese der sequentiell unabhängigen Ableitungen)
Sei $\rho = (G \overset{q'}{\Rightarrow} X \overset{q''}{\Rightarrow} H)$ eine sequentiell unabhängige Ableitung. Dann gibt es eine parallele direkte Ableitung $\rho' = (G \overset{q'+q''}{\Rightarrow} H)$, heißt eine *Synthese* von ρ. In diesem Fall schreiben wir $\rho' \in SYNT(\rho)$.
Bemerkung: Diese Ausführung ist im Allgemein nicht deteministisch.

Beispiel 4.10 (Analysis und Synthese)
Sei $\rho = (G_1 \overset{\texttt{SER}}{\Rightarrow} G_2 \overset{\texttt{REQ}}{\Rightarrow} G_3)$ die Ableitung in der Abbildung 4.3 und $\rho' = (G_1 \overset{\texttt{SER+REQ}}{\Longrightarrow} G_3)$ die parallele direkte Ableitung in der Abbildung 4.5. Weil ρ sequentiell unabhängig ist(siehe Beispiel 3.3), gibt es eine parallele direkte Ableitung, z.B. ρ'. Also haben wir $\rho' \in SYNT(\rho)$. Ähnlich haben wir $\rho \in ANAL_{\langle 1 \rangle, \langle 2 \rangle}(\rho')$. Jetzt sei $\rho'' = (G_1 \overset{\texttt{REQ}}{\Rightarrow} G_2' \overset{\texttt{SER}}{\Rightarrow} G_3)$ die Ableitung in der folgende Abbildung. Dann haben wir $\rho'' \in ANAL_{\langle 1 \rangle, \langle 2 \rangle}(\rho')$.

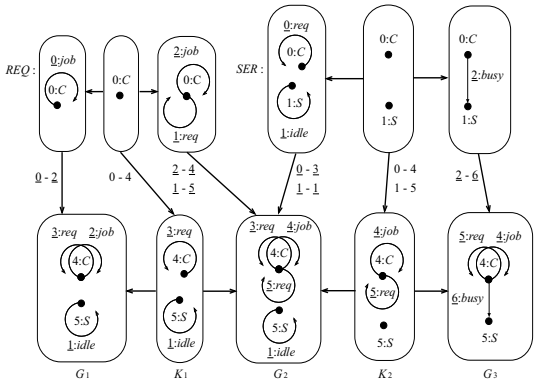


Abbildung 4.7 (a)

Jetzt können wir das Problem 3.2 im Abschnitt 3 lösen. Nach Lemma 4.8 ist die *Sequentialization-Kondition* Teil der Gluing-Kondition der parallelen Produktion, d.h, jede parallele direkte Ableitung kann in eine Sequenz der sequentiell unabhängigen direkten Ableitungen transformiert werden. Nach Lemma 4.9 ist die *Parallelization-Kondition* genau die sequentielle Unabhängigkeit der Ableitung, d.h, jede sequentiell unabhängige Ableitung kann in eine parallelen direkten Ableitung transformiert werden.

Theorem 4.11 (Parallelität)
Seien die Produktionen $p_1 : (L_1 \overset{l_1}{\leftarrow} K_1 \overset{r_1}{\rightarrow} R_1)$ und $p_2 : (L_2 \overset{l_2}{\leftarrow} K_2 \overset{r_2}{\rightarrow} R_2)$ gegeben. Dann sind die folgenden Aussagen äquivalent:

1. Es gibt eine parallele direkte Ableitung $G \overset{p_1+p_2,m}{\Longrightarrow} X$
2. Es gibt eine sequentiell unabhängige Ableitung $G \overset{p_1,m_1}{\Longrightarrow} H_1 \overset{p_2,m_2}{\Longrightarrow} X$.

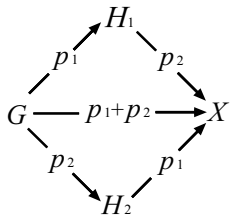


Abbildung 4.8

Am Ende dieses Abschnitts werden wir mit Hilfe der obigen Sätze beweisen, dass die parallele und sequenzielle Unabhängigkeit genau die gesuchten Bedingungen im Local Church-Rosser Problem (Problem 3.2) im Abschnitt 3.2 sind.

Theorem 4.12 (Local Church Rosser)
1. Seien $G \overset{p_1,m_1}{\Longrightarrow} H_1$ und $G \overset{p_2,m_2}{\Longrightarrow} H_2$ zwei parallele unabhängige direkte Ableitungen wie in der Abbildung 4.1. Ferner sei $m_2' = r_{1^* \circ k_1}$. Dann gibt es eine direkte Ableitung $H_1 \overset{p_2,m_2'}{\Longrightarrow} X$ und weiterhin ist die Transformationssequenz $G \overset{p_1,m_1}{\Longrightarrow} H_1 \overset{p_2,m_2'}{\Longrightarrow} X$ sequentiell unabhängig.

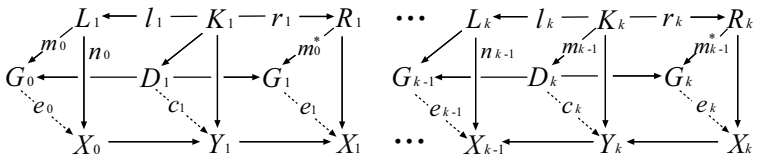
2. Sei $G \overset{p_1,m_1}{\Longrightarrow} H_1 \overset{p_2,m_2}{\Longrightarrow} H_2$ eine sequentiell unabhängige Ableitung wie in der Abbildung 4.2. Ferner sei $m_2' = l_{1^* \circ k_1}$. Dann gibt es eine direkte Ableitung $G \overset{p_2,m_2'}{\Longrightarrow} Y$ und weiterhin sind die direkten Ableitungen $G \overset{p_1,m_1}{\Longrightarrow} H_1$ und $G \overset{p_2,m_2'}{\Longrightarrow} Y$ parallel unabhängig.

Beweis:$G \overset{p_1,m_1}{\Longrightarrow} H_1$ und $G \overset{p_2,m_2}{\Longrightarrow} H_2$ sind zwei parallele unabhängige direkte Ableitungen, genau dann wenn (nach Satz 4.7) es eine Graph H existiert, so dass $G \overset{p_1+p_2,[m_1,m_2]}{\Longrightarrow} H$ eine parallele direkte Ableitung ist, genau dann wenn (nach Theorem 3.4.8) es eine sequentiell unabhängige Ableitung $G \overset{p_1,m_1}{\Longrightarrow} H_1 \overset{p_2,m_2'}{\Longrightarrow} X$ gibt. Somit sind die beiden Aussagen gezeigt.

4 Einbettung, Vereinigung und Verteilung für den DPO Ansatz

4.1 Einbettung der Ableitungen und abgeleitete Produktionen

Definition 5.1 (Einbettung und direkte abgeleitete Produktionen)
Seien $\rho = (G_0 \overset{p_1,m_0}{\Longrightarrow} \dots \overset{p_k,m_{k-1}}{\Longrightarrow} G_k)$ und $\delta = (X_0 \overset{p_1,n_0}{\Longrightarrow} \dots \overset{p_k,n_{k-1}}{\Longrightarrow} X_k)$ Ableitungen. Dann ist eine *Einbettung von ρ auf δ*, bez.: $\rho \overset{e}{\to} \delta$, eine Familie der Injektionen $e = (G_i \overset{e_i}{\to} X_i)_{i\in\{1,\dots,k\}}$, so dass für alle $i \in \{1, \dots, k\}$ gilt:
• $e_i \circ m_i = n_i$, und
• es gibt eine Injektion $D_i \overset{c_i}{\to} Y_i$, so dass das folgende Diagramm kommutiert:



Abbildung

Für gegebene ρ und δ ist die Einbettung e eindeutig bestimmt durch die erste Injektion e_0. So heißt e_0 *die Einbettungsmorphismus* von e. Sei $d = (G \overset{p,m}{\Longrightarrow} H)$ eine direkte Ableitung. Dann bezeichnen wir $\langle d\rangle : (G \overset{l^*}{\longleftarrow} D \overset{r^*}{\longrightarrow} H)$ *die direkte abgeleitete Produktion von d*, wobei $G \overset{l^*}{\longleftarrow} D \overset{r^*}{\longrightarrow} H$ die untere Zeile des Double-Pushout Diagramms (1)+(2) in der Abbildung 3.17 (l) ist. Eine direkte Ableitung $X \overset{\langle d\rangle,e}{\Longrightarrow} Z$ heißt eine *direkte abgeleitete Ableitung.*

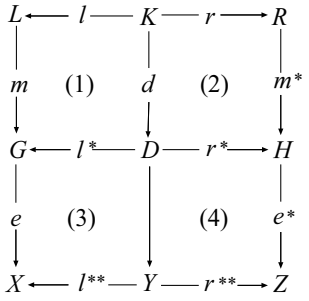


Abbildung 3.17(l)

Aus der Abbildung 3.17(l) bekommen wir einfach die foldende Satz.

Satz 5.2 (Direkte abgeleitete Ableitung)
Sei $\langle d\rangle$ die direkte abgelteitete Produktion einer Ableitung $d = (G \overset{p,m}{\Longrightarrow} H)$ wie in der

Abbildung 3.17. Dann sind die folgende Aussagen zueinander äquivalent:
1. Es gibt eine direkte abgeleitete Ableitung $X \overset{\langle d\rangle, e}{\Longrightarrow} Z$.
2. Es gibt eine direkte Ableitung $X \overset{p, e\circ m}{\Longrightarrow} Z$.

Definition 5.3 (Sequentiell zusammengesetzte Produktion)
Seien $p_1 : (L_1 \overset{l_1}{\longleftarrow} K_1 \overset{r_1}{\longrightarrow} R_1)$ und $p_2 : (L_2 \overset{l_2}{\longleftarrow} K_2 \overset{r_2}{\longrightarrow} R_2)$ mit $R_1 = L_2$ zwei Produktionen. Dann bezeichnen wir $p_1; p_2 : (L_1 \overset{l_1\circ l_2'}{\longleftarrow} K \overset{r_2\circ r_1'}{\longrightarrow} R_2)$ *eine sequentiell zusammengesetzte Produktion*, wobei $L_1 \overset{l_1\circ l_2'}{\longleftarrow} K \overset{r_2\circ r_1'}{\longrightarrow} R_2$ die obere Zeile des Diagramms in der Abbildung 3.17(r) ist.

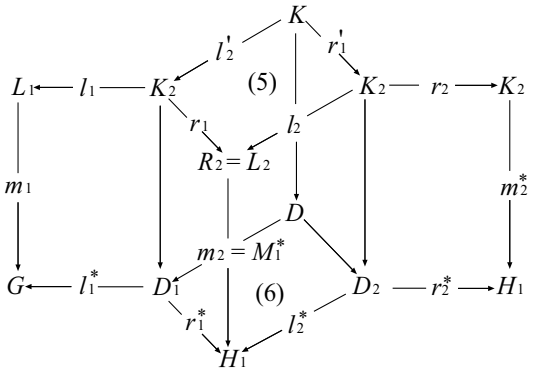


Abbildung 3.17(r)

Analogerweise bekommen wir aus der Abbildung 3.17(r) die folgenden äquivalenten Aussagen.

Satz 5.4 (Sequentielle Komposition)
Seien p_1, p_2 zwei Ableitungen und seine sequentiell zusammengesetzte Produktion $p_1; p_2$ wie in der Definition 3.5.3. Dann sind die folgenden Aussagen zueinander äquivalent:
1. Es gibt eine direkte Ableitung $G \overset{p_1;p_2, m_1}{\Longrightarrow} H_2$.
2. Es gibt eine Ableitung $G \overset{p_1, m_1}{\longleftarrow} H_1 \overset{p_2, m_1^*}{\longrightarrow} H_2$, so dass m_1^* das Ko-Matching von m_1 mit p_1 (wie in der Definition von direkte Ableitung).

Definition 5.5 (Abgeleitete Produktionen)
Die abgeleitete Produktion $\langle\rho\rangle$ *von der Ableitung* $\rho = (G_0 \overset{p_1, m_0}{\Longrightarrow} \ldots \overset{p_k, m_{k-1}}{\Longrightarrow} G_k)$ ist definiert durch die sequentielle Komposition $\langle d_1\rangle; \ldots; \langle d_k\rangle$, wobei $\langle d_i\rangle$ die direkte abgeleitete Produktion von $d_i = (G_{i-1} \overset{p_i, m_{i-1}}{\Longrightarrow} G_i)$ von ρ. Eine direkte Ableitung $K \overset{\langle\rho\rangle, e}{\Longrightarrow} Y$ heißt *abgeleitete Ableitung*.

Nach obigen Sätzen und Notationen können wir jetzt das Problem 3.2.4 im Abschnitt 3.2.3 lösen: $\langle\rho\rangle$ ist genau die gesuchte abgeleitete Produktion von ρ.

Theorem 5.6 (Abgeleitete Produktionen)
Sei ρ eine Ableitung und $\langle\rho\rangle$ seine abgeleitete Produktion wie in der Definition 3.5.5. Dann sind die folgenden Aussagen zueinander äquivalent:
1. Es gibt eine abgeleitete Ableitung $X_0 \overset{\langle\rho\rangle, e_0}{\Longrightarrow} X_k$.
2. Es gibt eine Ableitung $\delta = (X_0 \overset{p_1,n_0}{\Longrightarrow} \ldots \overset{p_k,n_{k-1}}{\Longrightarrow} X_k)$ und eine Einbettung $\rho \overset{e}{\rightarrow} \delta$, wobei e_0 die Einbettungsmorphismus von e ist.

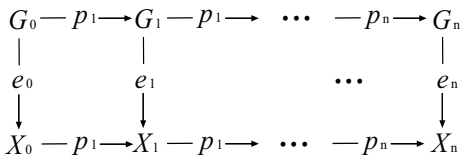


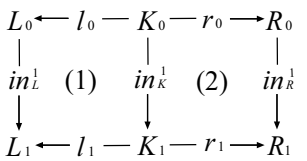
Abbildung S.176

Nach Theorem 3.5.6 bekommen wir auch die Lösung zum Problem 3.2.3 im Abschnitt 3.2.3.

Theorem 5.7 (Einbettung)
Sei $\rho = (G_0 \overset{p_1,m_0}{\Longrightarrow} \ldots \overset{p_k,m_{k-1}}{\Longrightarrow} G_k)$ eine Ableitung, $\langle\rho\rangle$ seine abgeleitete Produktion und $G_0 \overset{e_0}{\rightarrow} X_0$ eine Einbettungsmorphismus. Es gibt eine Ableitung $\delta = (X_0 \overset{p_1,e_0 \circ m_0}{\Longrightarrow} \ldots \overset{p_k,e_{k-1} \circ m_{k-1}}{\Longrightarrow} X_k)$ und eine Einbettung $\rho \overset{e}{\rightarrow} \delta$, genau dann wenn $\langle\rho\rangle$ anwendbar in e_0 ist, d.h. es gibt eine abgeleitete Ableitung $X_0 \overset{\langle\rho\rangle, e_0}{\Longrightarrow} X_k$.

4.2 Vereinigungen und Verteilungen

Definition 5.8 (Vereinigte Produktionen)
Seien $p_i : (L_i \overset{l_i}{\longleftarrow} K_i \overset{r_i}{\longrightarrow} R_i)$ drei Produktionen für $i \in \{0, 1, 2\}$. Dann heißt die Produktion p_0 zusammen mit den Graphmorphismen $in_L^1 : L_0 \rightarrow L_1$, $in_K^1 : K_0 \rightarrow K_1$ und $in_R^1 : R_0 \rightarrow R_1$ eine *Subproduktion von* p_1, wenn das folgende Diagramm kommutiert. Die Einbettung von p_0 zu p_1 wird bezeichnet als $in^1 = \langle in_L^1, in_k^1, in_R^1\rangle : p_0 \rightarrow p_1$.



Abbildung

Wenn p_0 Subproduktionen von p_1 und p_2 mit Einbettungen in^1 und in^2 sind, sagen wir: Produktionen p_1 und p_2 sind *synchronisiert*, bez. : $p_1 \xleftarrow{in^1} p_0 \xrightarrow{in^2} p_2$. In diesem Fall heißt $p_1 \oplus_{p_0} p_2 : (L \xleftarrow{l} K \xrightarrow{r} R)$ im folgenden Diagramm *die vereinigte Produktion*, wenn die Subdiagramme (1),(2) und (3) Pushouts sind und alle Diagramme kommutieren. Eine direkte Ableitung $G \overset{p,m}{\Longrightarrow} X$, wobei $p = p_1 \oplus_{p_0} p_2$, heißt *vereinigte Ableitung*.

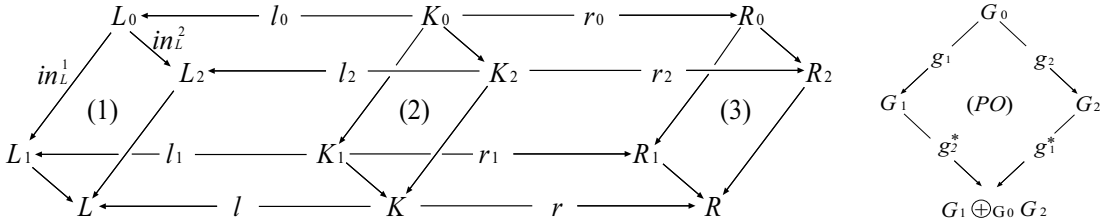


Abbildung P225

Definition 5.9 (Verteilte Graphen)
Ein *verteilter Graph* $DG = (G_1 \xleftarrow{g_1} G_0 \xrightarrow{g_2} G_2)$ enthält zwei *lokale Graphen* G_1 und G_2, einen *Interfacegraph* G_0, und Graphmorphismen g_1 und g_2, die den Interfachgraph in die lokalen Graphen einbetten. Der globale Graph $\oplus DG = G_1 \oplus_{G_0} G_2$ von DG ist definiert durch die Pushout-Objekte von g_1 und g_2, wie in der folgende Abbildung. DG heißt eine *Spaltung* von $\oplus DG$.

Die lokale Graphen können natürlich mit lokalen direkten Ableitungen transformiert werden. Um *verteilte Ableitungen* zu formulieren, brauchen wir noch das *verteilte Matching* zu definieren.

Definition 5.10 (Verteilte Matching)
Sei DG ein verteilter Graph, $p_1 \xleftarrow{in^1} p_0 \xrightarrow{in^2} p_2$ synchronisierte Produktionen, und $L_i \xrightarrow{m_i} G_i$, $(i \in \{0,1,2\})$ Matchings für p_i, wobei $g_k \circ m_0 = m_k \circ in_L^k$ für $k \in \{1,2\}$. In diesem Fall heißt die Familie $(m_i)_{i\in\{0,1,2\}}$ ein *verteiltes Matching* für $p_1 \xleftarrow{in^1} p_0 \xrightarrow{in^2} p_2$ in DG. Die *verteilte Gluing-Kondition* wird erfüllt, wenn alle m_i die lokale Gluing-Kondition von p_i und die *Connection-Kondition* erfüllen, d.h., $g_k(G_0 - m_0(L_0 - l_0(K_0))) \subseteq G_k - m_k(L_k - l_k(K_k))$ für $k \in \{1,2\}$.

Definition 5.11 (Verteilte Ableitungen)
Sei $(m_i)_{i\in\{0,1,2\}}$ ein verteiltes Matching wie in der Definition 3.5.10. Es gibt lokale direkte Ableitungen $d_i = (G_i \overset{p_i,m_i}{\Longrightarrow} H_i)$ für $i \in \{0,1,2\}$ und verteilte Graphen $DG = (G_1 \xleftarrow{g_1} G_0 \xrightarrow{g_2} G_2)$, $DD = (D_1 \xleftarrow{c_1} D_0 \xrightarrow{c_2} D_2)$ und $DH = (H_1 \xleftarrow{h_1} H_0 \xrightarrow{h_2} H_2)$, so dass $d_1 \|_{d_0} d_2 : DG \Rightarrow DH$ eine *verteilte Ableitung* bildet, wie in der Abbildung 3.18. Die Graphmorphismen c_1 und c_2 von dem verteilten Graph DD sind so

definiert: $c_1 = l_1^{*-1} \circ g_1 \circ l_0^*$ und $c_2 = l_2^{*-1} \circ g_2 \circ l_0^*$. h_1 und h_2 von DH sind entsprechend.

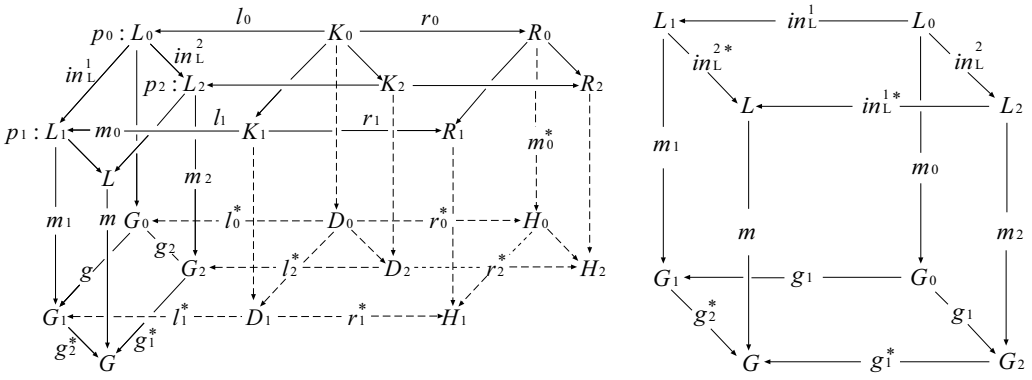


Abbildung 3.18

Die folgende Theorem gibt eine Lösung zu dem Problem 3.2.5 im Abschnitt 3.2.4 aus.

Theorem 5.12 (Verteilung)
Sei DG ein verteilter Graph und $p_1 \xleftarrow{in^1} p_0 \xrightarrow{in^2} p_2$ synchronisierte Produktionen. Dann sind die folgenden Aussagen zueinander äquivalent:
1. Es gibt eine vereinigte Ableitung $\oplus DG \overset{p_1 \oplus_{p_0} p_2, m}{\Longrightarrow} H$ und ein verteiltes Matching $(m_i)_{i \in \{0,1,2\}}$, so dass
(a) $g_2^* \circ m_1 = m \circ in_L^{2*}$ und $g_1^* \circ m_2 = m \circ in_L^{1*}$, d.h., mit der Voraussetzung des verteilten Matchings (also $g_k \circ m_0 = m_k \circ in_L^k$ für $k \in \{1,2\}$) kommutiert das rechte Diagramm in der Abbildung 3.18.
(b) $(m_i)_{i \in \{0,1,2\}}$ wird die Connection-Kondition erfüllt (siehe Definition 3.5.10), und
(c) für $i = 0,1,2$, $m_i|^{S_i} : L_i \to S_i$ wird die lokale Gluing-Kondition von p_i erfüllt, wobei $S_1 = g_2^{*-1}(m(L))$, $S_2 = g_1^{*-1}(m(L))$, $S_0 = (g_2^* \circ g_1)^{-1}(m(L))$, und $m_i|^{S_i}$ ist m_i eingeschränkt auf S_i.

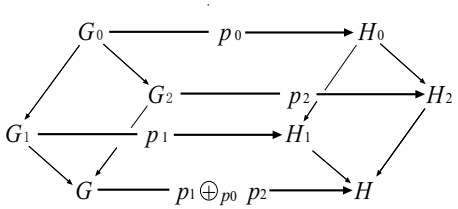


Abbildung P180

2. Es gibt ein verteilter Graph DH mit $\oplus DH = H$ und eine verteilte Ableitung $d_1 \|_{d_0} d_2 : DG \Rightarrow DH$ mit $d_i = (G_i \overset{p_i, m_i}{\Longrightarrow} H_i)$ für $i = 0,1,2$.

Literatur

[1] *Handbook of Graph Grammars and Computing by Graph Transformation, Volume 1 - Kapitel 3* ,