Verhaltensäquivalenzen für CCS

Holger Dell
Lehrstuhl für Programmiersysteme - Prof. Gert Smolka

18. Oktober 2004

Nachdem nun CCS-Ausdrücke eingeführt sind, wendet sich Robin Milner in Kapitel 5 und 6 von "Communicating and Mobile Systems: the π-calculus" möglichen Verhaltensäquivalenzen über CCS zu: die über Bisimulationen definierte starke bzw. schwache Äquivalenz.

Inhaltsverzeichnis

1 Einführung

1.1 CCS-Semantik

Wir erinnern uns an die Definition von CCS-Prozessausdrücken:

$$P ::= A(a_1, \cdots, a_n) \mid \sum_{i \in I} \alpha_i P_i \mid (P_1 | P_2) \mid \text{new } \alpha P$$

Die Menge aller Prozessausdrücke war mit $\mathcal{P}$ bezeichnet. Im Folgenden wird Milner formalisieren, was es bedeutet, dass zwei Prozesse $P_1, P_2 \in \mathcal{P}$ gleiches Verhalten zeigen.
Dazu nutzt er die am Anfang von Kapitel 5 eingeführte operationale Semantik $\xrightarrow{\alpha}$, die über $\mathcal{P}$ definiert ist. Durch sie kann jeder CCS-Ausdruck $P \in \mathcal{P}$ als Zustand eines Transitionssystems (LTS = *labelled transition system*) angesehen werden. Jeder von P durch Transitionen erreichbare Prozessausdruck P' wird dabei in die (möglicherweise undendlich große) Zustandsmenge mit aufgenommen.
Milner bedient sich bei der Definition der Äquivalenzen also *nicht* der in Kapitel 4 mit den *reaction rules* eingeführten anschaulicheren Semantik der *chemical abstract machine* von Berry und Boudol (1990), zeigt aber zumindest, dass diese beiden Semantiken äquivalent sind ($P \xrightarrow{\tau} \equiv P' \Leftrightarrow P \to P'$).

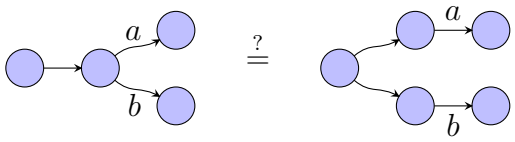
1.2 Beobachtbarkeit

In der Physik gibt es das große Feld der Beobachtungstheorie, die beschreibt, welche Dinge überhaupt beobachtbar sind oder wie Dinge beobachtet werden sollen oder können. Der Physiker bedient sich beispielsweise eines Mikroskops, um kleine Dinge wahrnehmen zu können, oder einer (Licht-)Blende, um bestimmte Dinge auszublenden.
Wenn wir nun über CCS-Prozessen Verhaltensäquivalenzen definieren wollen, müssen wir uns vor Augen halten, welche Dinge wir sehen wollen und welche Dinge wir ausblenden wollen.
Wir transformiern die CCS-Ausdrücke mit Hilfe der operationalen Semantik $\xrightarrow{\alpha}$ in Transitionssysteme (LTS) und blenden so die konkrete Syntax der CCS-Ausdrücke aus, da wir jetzt nur noch abstrakte Graphen betrachten.

1.3 Unterscheidung von CCS-Ausdrücken

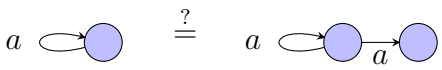
In der klassischen Automatentheorie definiert man Äquivalenz von solchen Graphen bzw. Automaten über die Gleichheit der von ihnen akzeptierten Sprachen. Dies reicht für CCS-Ausdrücke aber nicht aus, da wir wollen, dass auch folgende Phänomene bei ihrer Unterscheidung berücksichtigt werden:

- Der Automat könnte an einen Punkt kommen, an dem er sich nichtdeterministisch für einen weiteren Verlauf entscheiden muss:

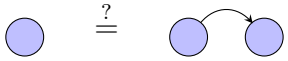
  Hierbei sind die Sprachen der beiden Automaten gleich ($\mathcal{L} = \{a, b\}$). Da der rechte Automat sich aber schon vorher entscheidet, ob der Benutzer (oder ein anderer Prozess) dem Automaten a oder b zum Lesen wird geben können, der linke Automat diese Wahl aber dem Benutzer selbst überlässt, werden wir diese beiden Prozesse nicht als äquivalent ansehen wollen. (Dies ist eine vereinfachte Variante der *vending machine*, die Milner als Beispiel erwähnt)

- Der Automat könnte nichtdeterministisch in einen *deadlock* geraten:

  Auch hier sind die Sprachen gleich ($\mathcal{L} = a^*$), aber der rechte Automat kann in einen *deadlock* geraten, während der linke weiterlaufen wird.

- Ein Automat könnte interne Arbeiten erledigen, die man von außen unter Umständen nicht sehen möchte:

  Der linke Automat macht gar nichts, während der rechte interne Arbeiten erledigt und dann ebenfalls nichts macht. Wir werden sehen, dass solche internen Vorgänge bei der starken Äquivalenz beobachtbar, bei der schwachen dagegen nicht beobachtbar sind.

2 starke Äquivalenz

2.1 Definition

Der Begriff der starken Äquivalenz ist schon in Kapitel 3 definiert worden, wir übertragen ihn hier lediglich auf durch die operationale Semantik $\xrightarrow{\alpha}$ induzierte Transitionsssteme.

Zur Erinnerung:

- Definition: eine Relation S heißt Simulation genau dann, wenn für alle Knoten $P, P', R \in \mathcal{P}$ und Aktionen α mit

  $$\begin{array}{ccc} P & \overset{\mathcal{S}}{\text{———}} & P' \\ \alpha\downarrow & & \\ R & & \end{array}$$ ein $R' \in \mathcal{P}$ existiert, sodass $$\begin{array}{ccc} & & P' \\ & & \downarrow\alpha \\ R & \overset{\mathcal{S}}{\text{———}} & R' \end{array}$$ gilt.

- Definition: $\mathcal{S}$ heißt Bisimulation genau dann, wenn $\mathcal{S}$ und $\mathcal{S}^{-1}$ Simulationen sind.

- Definition: P heißt stark äquivalent zu P' ($P \sim P'$) genau dann, wenn es eine Bisimulation $\mathcal{S}$ gibt mit $P\mathcal{S}P'$.

2.2 Eigenschaften

Ein paar interessante Eigenschaften von $\sim$ sind noch einmal hervorgehoben:

- Wie man leicht durch Widerspruch zeigen kann ist $\sim$ ist die größte Bisimulation.

- Aus Theorem 5.13 (2) ($P \equiv Q \Rightarrow P \sim Q$) ergibt sich, dass die Äquivalenzklassen von $\sim$ über $\mathcal{P}$ mindestens so viele Elemente enthalten, wie die von $\equiv$. Das heißt, die starke Äquivalenz ist in diesem Sinne schwächer als die strukturelle Kongruenz.

- Es gilt $(\text{new}a)a.P = 0 = (\text{new}a)\overline{a}.P$ (siehe Beispiel 5.26)

- $\sim$ ist eine *process congruence* (siehe Proposition 5.29); gleiches darf also für gleiches ersetzt werden.

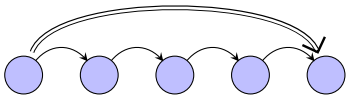
- Um $P' \sim P$ zu testen, genügt es bei der Definition der Simulation, eine Relation $\mathcal{S}$ zu finden, sodass für alle Knoten P, P', R und Aktionen α mit

  $$\begin{array}{ccc} P & \overset{\mathcal{S}}{\text{———}} & P' \\ {\scriptstyle\alpha}\downarrow & & \\ R & & \end{array}$$ ein $R' \in \mathcal{P}$ existiert, sodass $$\begin{array}{ccc} & & P' \\ & & \downarrow{\scriptstyle\alpha} \\ R & \overset{\equiv \mathcal{S} \equiv}{\text{———}} & R' \end{array}$$ gilt.

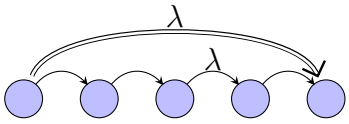
3 schwache Äquivalenz

3.1 Definition

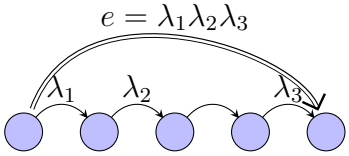
Wie bereits erwähnt wollen wir τ-Transitionen, also interne Vorgänge bei der schwachen Äuivalenz nicht beobachten. Also liegt es nahe, die Definition der Simulation dahingehend zu überdenken, ob tatsächlich jede τ-Transition im einen Prozess auch im andern Prozess vorhanden sein muss. Wir werden daher eine neue Transititonsrelation definieren, die τ einfach "überliest", also nicht beobachtet.



$$\Rightarrow \quad := \quad \rightarrow^*$$



$$\stackrel{\lambda}{\Rightarrow} \quad := \quad \Rightarrow \stackrel{\lambda}{\rightarrow} \Rightarrow$$



$$\stackrel{e}{\Rightarrow} \quad := \quad \stackrel{\lambda_1}{\Rightarrow}\stackrel{\lambda_2}{\Rightarrow}\stackrel{\lambda_3}{\Rightarrow} \ldots$$

Nun sind wir in der Lage, die Definition der schwachen Äquivalenz mit Hilfe von $\stackrel{e}{\Rightarrow}$ analog zu der starken Äquivalenz zu geben.

- Definition: eine Relation S heißt schwache Simulation genau dann, wenn für alle Knoten $P, P', R \in \mathcal{P}$ und Experimente e mit

  $$\begin{array}{ccc} P & \overset{\mathcal{S}}{\text{———}} & P' \\ e\Downarrow & & \\ R & & \end{array}$$ ein $R' \in \mathcal{P}$ existiert, sodass $$\begin{array}{ccc} & & P' \\ & & \Downarrow e \\ R & \overset{\mathcal{S}}{\text{———}} & R' \end{array}$$ gilt.

  Diese Definition ist zwar analog zu der Definition der starken Simulation, für Beweise allerdings unhandlich, da immer alle möglichen Experimente e betrachtet werden müssen, und das können mitunter unendlich viele sein.

- Satz: eine Relation S ist eine schwache Simulation genau dann, wenn für alle Knoten $P, P', R \in \mathcal{P}$ und Aktionen α mit

  $$\begin{array}{ccc} P & \overset{\mathcal{S}}{\text{---}} & P' \\ {\scriptstyle\alpha}\downarrow & & \\ R & & \end{array}$$ ein $R' \in \mathcal{P}$ existiert, sodass $$\begin{array}{ccc} & & P' \\ & & \Downarrow\!\alpha \\ R & \overset{\mathcal{S}}{\text{---}} & R' \end{array}$$ gilt.

  Nun muss man immer nur für alle solche Knoten die endlich vielen Transitionen $\xrightarrow{\alpha}$ überprüfen, die von ihnen wegführen.

- Definition: $\mathcal{S}$ heißt schwache Bisimulation genau dann, wenn $\mathcal{S}$ und $\mathcal{S}^{-1}$ schwache Simulationen sind.

- Definition: P heißt schwach äquivalent zu P' ($P \approx P'$) genau dann, wenn es eine schwache Bisimulation $\mathcal{S}$ gibt mit $P\mathcal{S}P'$.

3.2 Eigenschaften

Ein paar interessante Eigenschaften von $\approx$ sind noch einmal hervorgehoben:

- Auch hier lässt sich leicht durch Widerspruch zeigen, dass $\approx$ die größte schwache Bisimulation.

- Aus Proposition 6.6 ($P \sim Q \Rightarrow P \approx Q$) ergibt sich, dass die Äquivalenzklassen von $\approx$ über $\mathcal{P}$ mindestens so viele Elemente enthalten, wie die von $\sim$. Das heißt, die schwache Äquivalenz ist in diesem Sinne schwächer als die starke Äquivalenz, was die Namenswahl rechtfertigt.

- Das grundlegende Ergebnis der schwachen Äquivalenz ist wohl: $\tau.P \approx P$ (Theorem 6.15 (2))

- $\approx$ ist eine *process congruence* (siehe Proposition 6.17); gleiches darf also für gleiches ersetzt werden (bei der Summation aber, wie Milner selbst auch anmerkt, nur hinter einer Aktion ($a.P$)).

- Um $P' \approx P$ zu testen, genügt es bei der Definition der schwachen Simulation, eine Relation $\mathcal{S}$ zu finden, sodass für alle Knoten P, P', R und Aktionen α mit

  $$\begin{array}{ccc} P & \overset{\mathcal{S}}{\text{---}} & P' \\ {\scriptstyle\alpha}\downarrow & & \\ R & & \end{array}$$ ein $R' \in \mathcal{P}$ existiert, sodass $$\begin{array}{ccc} & & P' \\ & & \Downarrow\!\alpha \\ R & \overset{\sim\,\mathcal{S}\,\sim}{\text{---}} & R' \end{array}$$ gilt.

4 Referenzen

- R. Milner, “Communicating and Mobile Systems: the π-Calculus”, Cambridge University Press, 1999.

- H. Hermanns, “Verification” (Vorlesungsskript), Universität des Saarlandes, 2003.

- G. Berry und G. Boudol, “The Chemical Abstract Machine.” (Theoretical Computer Science), 96:217-248, 1992.