Réécriture

Complétion

version du November 29, 2004 – 15 h 27

Le lemme de Knuth-Bendix et la confluence locale

Paires critiques

Si il existe

- deux règles $l \longrightarrow r$ et $g \longrightarrow d$

  dont on suppose les ensembles de variables disjoints,

- une position $p \in Pos(l)$, qui n'est pas la position d'une variable,
- un mgu σ de $l_{|p}$ et de g.

Le terme $\sigma(l)$ est appelé une superposition

et la paire $\langle \sigma(r)\ ,\ \sigma(l[d]_p) \rangle$ est appelée une paire critique.

Paires critiques

Si il existe

- deux règles $l \longrightarrow r$ et $g \longrightarrow d$

  dont on suppose les ensembles de variables disjoints,

- une position $p \in Pos(l)$, qui n'est pas la position d'une variable,
- un mgu σ de $l_{|p}$ et de g.

Le terme $\sigma(l)$ est appelé une superposition

et la paire $\langle \sigma(r)\ ,\ \sigma(l[d]_p) \rangle$ est appelée une paire critique.

Le problème est de savoir si les paires critiques sont joignables.

Les positions non prises en compte

Position non variable

Si dans la définition, on acceptait une position $p \in Pos(l)$ qui est la position d'une variable x, alors

- il y aurait toujours un mgu de σ de $l_{|p}$ et de g, à savoir $\{x \mapsto g\}$,
- il y aurait donc toujours une paire issue de cette superposition (triviale), à savoir $\langle \{x \mapsto g\}(r)\ ,\ \langle \{x \mapsto g\} l[d]_p \rangle$,
- cette paire critique serait joignable vers le terme $\{x \mapsto d\}(r)$.

Donc on ne considère pas les positions dans l qui correspondent à des variables.

Les positions non prises en compte

Superposition à la racine dans le cas où $l \to r \equiv g \to d$.

Dans ce cas, la superposition est triviale, le mgu est la substitution identité et la paire critique est évidemment joignable.

On ne considère donc pas les superpositions à la racine d'une règle sur elle-même.

En revanche, d'autres superpositions peuvent exister comme dans le cas où

- $l \to r \equiv (x_1 * y_1) * z_1 \to x_1 * (y_1 * z_1)$
- et $g \to d \equiv (x_2 * y_2) * z_2 \to x_2 * (y_2 * z_2)$.

Un exemple

associativité + endomorphisme

$$(x * y) * z \quad \longrightarrow \quad x * (y * z) \tag{1}$$

$$f(x) * f(y) \quad \longrightarrow \quad f(x * y) \tag{2}$$

Paires Critiques

$$(f(x) * f(y)) * z \quad \xrightarrow{1} \quad f(x) * (f(y) * z)$$

$$\xrightarrow{2} \quad f(x * y) * z$$

$(f(x) * f(y)) * z$ est une superposition

de la règle (1) dans la règle (2).

$$\langle f(x) * (f(y) * z), f(x * y) * z \rangle$$

est une paire critique **non joignable**.

$((x_1 * x_2) * y) * z$ est une superposition

de la règle (2) dans la règle (2).

$$\begin{aligned} ((x_1 * x_2) * y) * z &\xrightarrow{2} (x_1 * (x_2 * y)) * z \\ &\xrightarrow{2} (x_1 * x_2) * (y * z) \end{aligned}$$

$$\langle (x_1 * (x_2 * y)) * z, (x_1 * x_2) * (y * z) \rangle$$

est une paire critique **joignable**, puisque les deux membres se réécrivent vers le même terme.

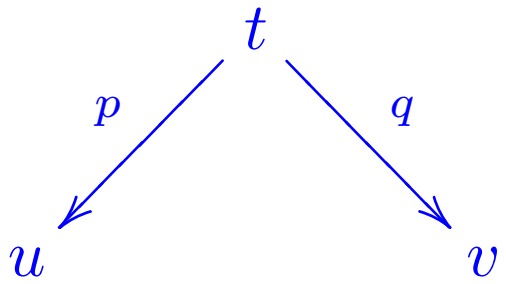
Théorème de Knuth et Bendix

Etant donné un système de réécriture R.

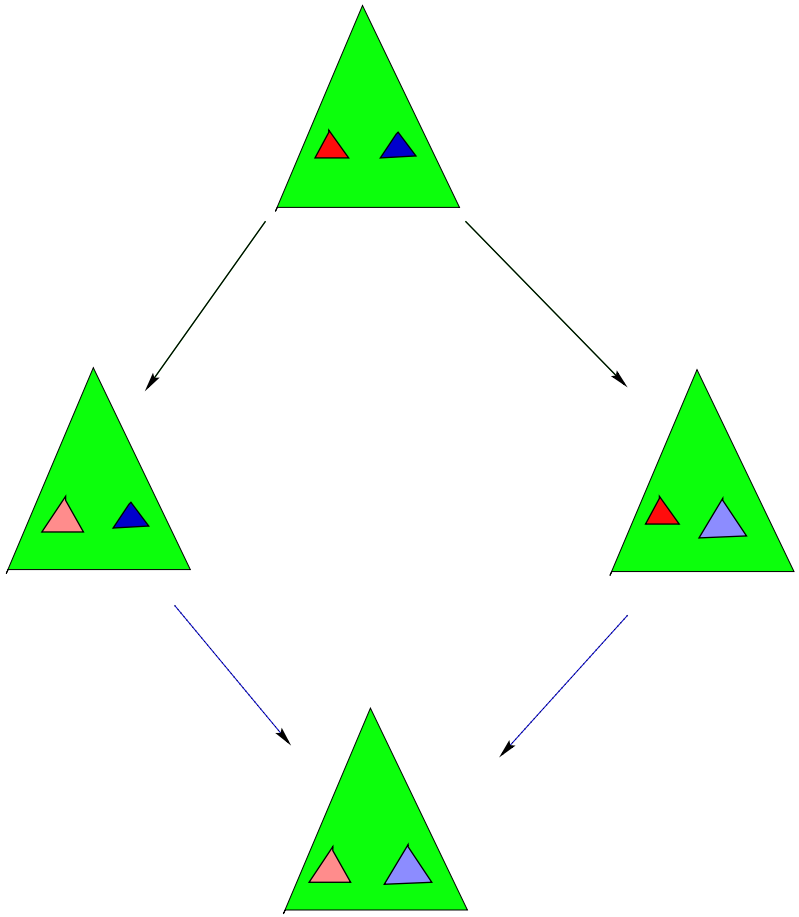
R est localement confluent si et seulement si toutes ses paires critiques sont joignables.

La condition est clairement nécessaire.

Pour la condition suffisante, il faut examiner les triplets



et considérer les positions respectives de p et q et les régles $l \longrightarrow r$ et $g \longrightarrow d$ qui permettent de réécrire t.

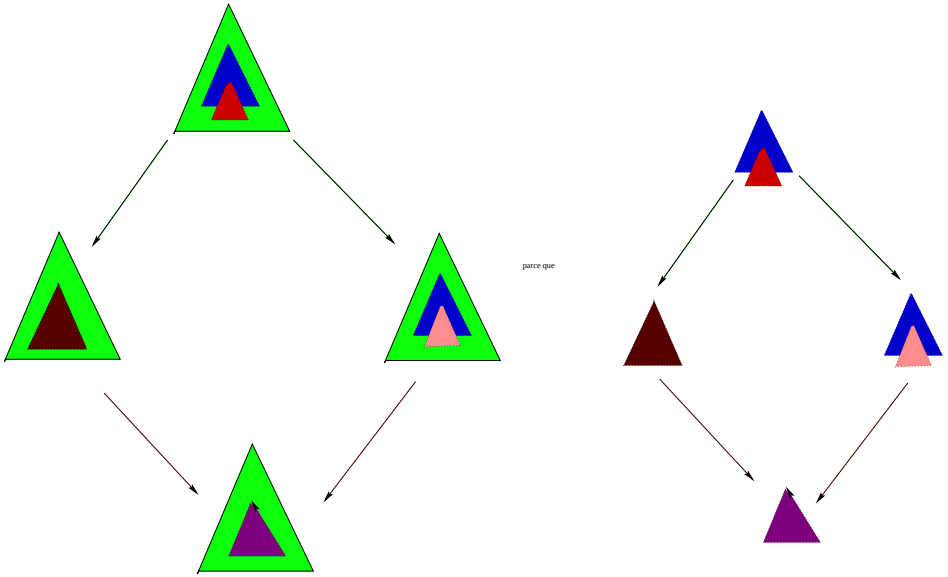


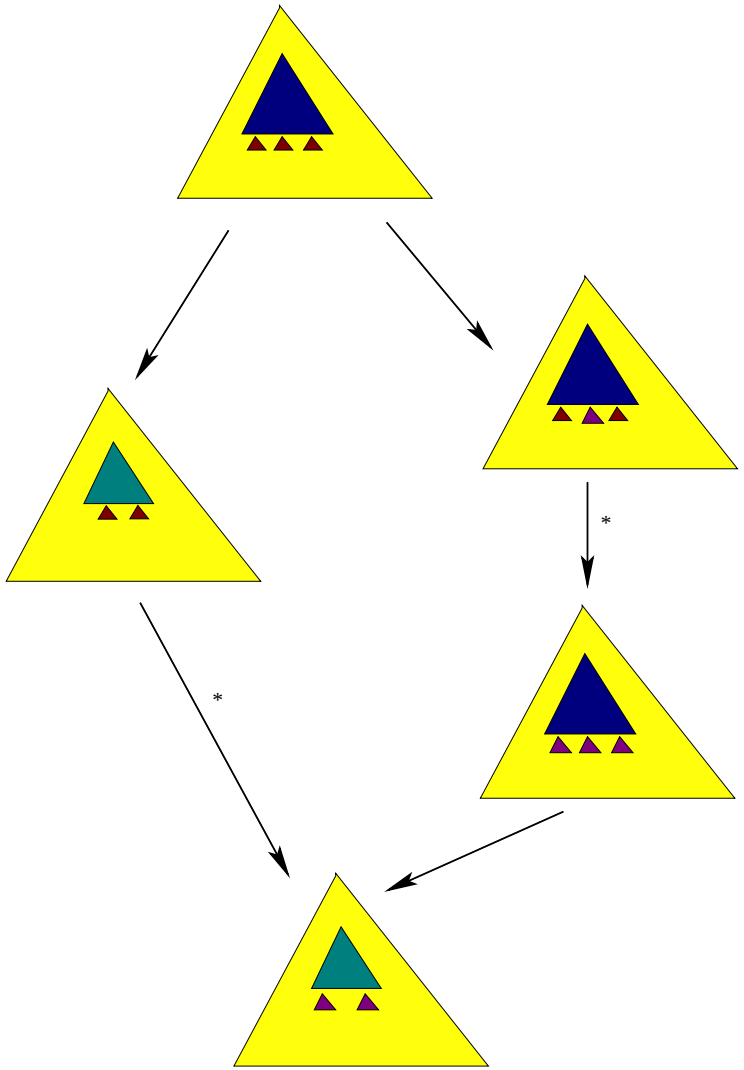
Si p et q sont étrangers alors

- $u \xrightarrow[g \to d, \rho]{q} u'$
- $v \xrightarrow[l \to r, \theta]{p} v'$,
- et $u' \equiv v'$.

Si p est un préfixe de q (avec $q \equiv pq'$) et $q' \in Pos(l)$, alors

- $t_{|p} = \theta(l)$ est l'instance d'une superposition,
- et $\langle \theta(r)\ ,\ \theta(l[d]_{q'} \rangle$, est l'instance d'une paire critique entre $l \to r$ et $g \to d$,
- on a le résultat parce que la paire critique est joignable.





Si p est un préfixe de q (avec $q \equiv pq'$) et $q' \notin Pos(l)$, on peut réécrire les “résidus”.

La compétion en Orme

Procédure de Complétion

Une procédure de complétion transforme un ensemble de d'identités en un ensemble équivalent de règles confluentes et qui se terminent.

Plus généralement, une procédure de complétion transforme

- un ensemble d'identités et de règles qui se terminent
- en un ensemble équivalent de règles confluentes et qui se terminent.

Cela nécessite un ordre pour orienter les identités.

Les composants d’ORME

Les composants en rouge sont des conteneurs qui ne contiennent qu’une règle, les autres sont des listes d’objets.

Les extensions **X_ext** ne sont pas utilisés.

Les composants d'ORME

AC_op les opérateurs associatifs et commutatifs,

C_op les opérateurs commutatifs,

E_raw les identités telles quelles,

E_norm les identités après normalisation par les règles du système,

S une règle qui vient d'être orientée sert à simplifi er,

T les règles en attente de traitement,

C la règle dont on calcule toutes les paires critiques,

N les règles dont la paire critique avec C n'a pas encore été calculée.

A les règles dont la paire critique avec C a déjà été calculée,

Quelques étapes d'ORME

New loop démarrage d'une nouvelle boucle de calcul de paires critiques,

Cleaning of E_raw nettoyage de **E_raw** pour créer **E_nor**,

Orientation of E_nor orientation d'une égalité normalisée,

Simplification utilisation de **S** pour simplifier les autres composants,

C_internal_Deduction calcul des paires critiques de **C** sur elle-même,

N_C_Deduction calcul des paires critiques de **C** avec celle de N.

Un exemple de complétion

La présentation des groupes de Taussky

L'exemple de Taussky

$*$ est associatif

$$x * (y * z) \quad = \quad (x * y) * z \qquad\qquad (A)$$

Il existe un élément idempotent:

$$e * e \quad = \quad e \qquad\qquad (Id)$$

Chaque élément a au moins un inverse à droit vis-à-vis de e:

$(\forall x)(\exists y)\ x * y = e$, ce qui donne par skolémisation:

$$x * i(x) \quad = \quad e \qquad\qquad (InvD)$$

L'exemple de Taussky

Chaque élément a au plus un inverse à gauche vis-à-vis de e:

$$(\forall x)\,(\forall x')\,(\forall y) \quad x*y = x'*y = e \quad \Rightarrow \quad x = x' \qquad\qquad (InvG)$$

ce qui est équivalent aux deux équations *(T)*:

$$\begin{aligned} g(x*y, y) &= f(x*y, x) \\ f(e, x) &= x. \end{aligned}$$

L'exemple de Taussky

(T) $\Rightarrow$*(InvG)*

Supposons $z' * z = z'' * z = e$.

$$\begin{aligned} z' &= f(e, z') = f(z' * z, z') = g(z' * z, z) \\ &= g(z'' * z, z) = f(z'' * z, z'') = f(e, z'') \\ &= z''. \end{aligned}$$

L'exemple de Taussky

(A) + *(Id)* + *(InvD)* + *(InvG)* $\Rightarrow$ *(T)*

On montre que l'ajout de (T) est conservatif autrement dit, si $s, t \in T(\{*, i, e), X)\}$ alors

$$(A) + (Id) + (InvD) + (InvG) + (T) \vdash s = t$$

$$\Rightarrow$$

$$(A) + (Id) + (InvD) + (InvG) \vdash s = t$$

En effet, *(T)* n'ajoute pas de nouvelles équations entre termes de $T(\{*, i, e), X)\}$, car une telle chaîne passerait par

$$s = f(e, s') = \ldots = g(e, u) \equiv g(s' * u, u) = g(t' * u, u) = \ldots = f(e, t')$$

Ce qui implique que $s = s' = t' = t$ puisque u a au plus un inverse à gauche vis-à-vis de e.

L'exemple de Taussky

On obtient les dix axiomes classiques plus les deux règles:

$$\begin{aligned} f(e,x) &\rightarrow x \\ g(x,y) &\rightarrow f(x, x * i(y)). \end{aligned}$$

Groupes (Axiomes de Taussky)

$$
\begin{aligned}
(x * y) * z &= x * (y * z) \\
e * e &= e \\
x * i(x) &= e \\
g(x * y, y) &= f(x * y, x) \\
f(e, x) &= x
\end{aligned}
$$

Groupes (Axiomes de Taussky, système complété)

$$
\begin{array}{rcl}
(x * y) * z & \longrightarrow & x * (y * z) \\
e * x & \longrightarrow & x \\
x * e & \longrightarrow & x \\
i(x) * x & \longrightarrow & e \\
x * i(x) & \longrightarrow & e \\
i(i(x)) & \longrightarrow & x \\
i(e) & \longrightarrow & e \\
i(x * y) & \longrightarrow & i(y) * i(x) \\
x * (i(x) * y) & \longrightarrow & y \\
i(x) * (x * y) & \longrightarrow & y \\
\hline
g(x, y) & \longrightarrow & f(x, i(y) * x) \\
f(e, x) & \longrightarrow & x
\end{array}
$$

La procédure de complétion

L'enchâssement

La relation d'enchâssement $\triangleright\!\bullet$ (en anglais encompassment) est définie par

$$s \triangleright\!\bullet\, t$$

si et seulement si

$$s \neq t \quad \text{et} \quad \exists p \in Pos(s) \cdot \exists \sigma \in Sub(T(\Sigma, V)) \cdot \quad s_{|p} = \sigma(t)$$

$\triangleright\!\bullet$ termine.

Delete: $E \cup \{s = s\}; R \quad \vdash \quad E; R$

Compose: $E; R \cup \{s \to t\} \quad \vdash \quad E; R \cup \{s \to u\}$

si $t \xrightarrow[R]{} u$

Simplify: $E \cup \{s = t\}; R \quad \vdash \quad E \cup \{s = u\}; R$

si $t \xrightarrow[R]{} u$

Collapse: $E; R \cup \{s \to t\} \quad \vdash \quad E \cup \{u = t\}; R$

si $s \xrightarrow[R]{} u$

par une règle $l \to r \in R$

avec $s \triangleright l$

ou $s = l$ et $t > r$

Orient: $E \cup \{s = t\}; R \quad \vdash \quad E; R \cup \{s \to t\}$

si $s > t$

Deduce: $E; R \quad \vdash \quad E \cup \{s = t\}; R$

si $s \xleftarrow[R]{} u \xrightarrow[R]{} t$

pour un u

Completion de l'*associativité + endomorphisme*

Commençons par

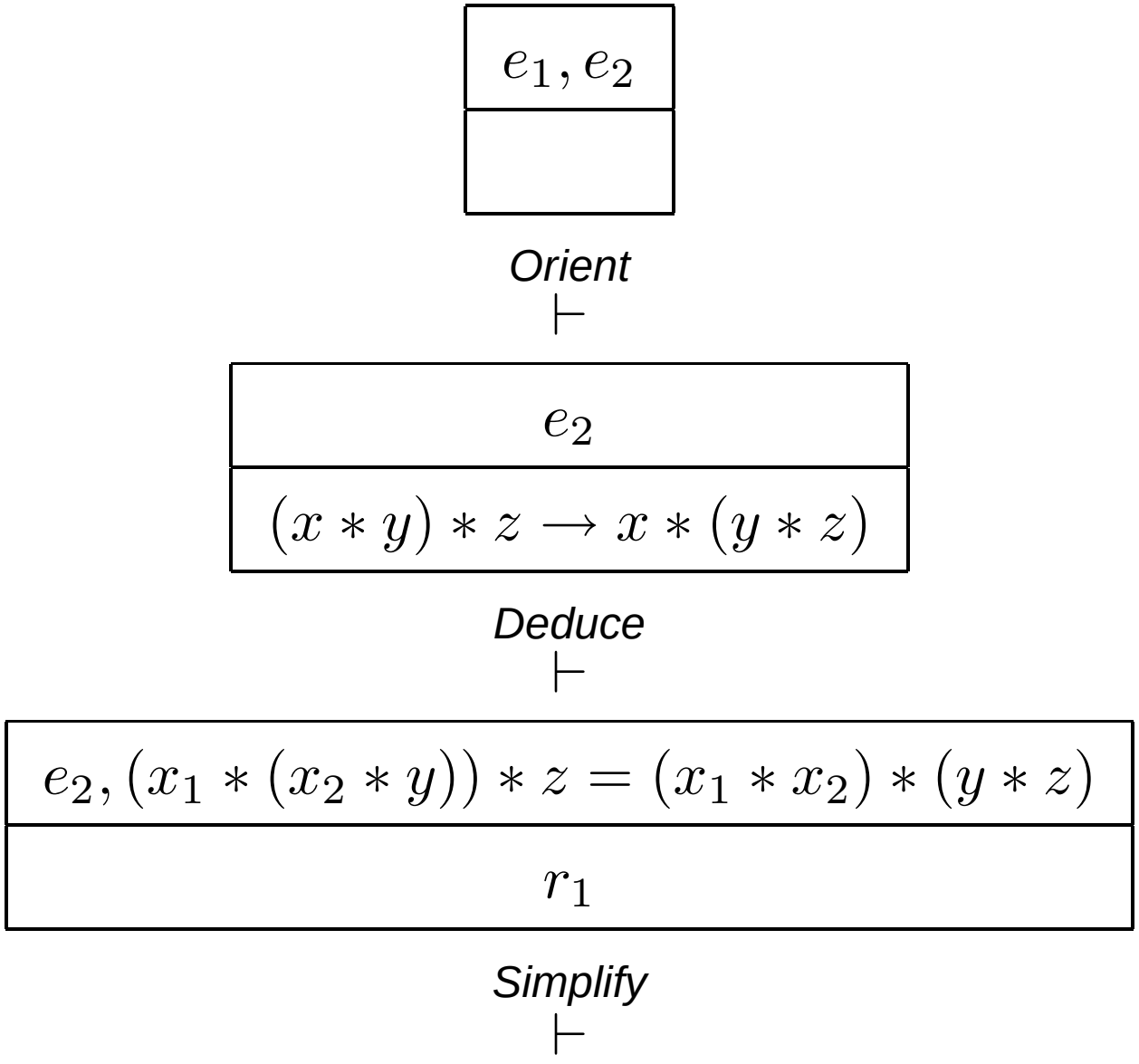
$$E_0 \equiv \{e_1, e_2\}$$

et

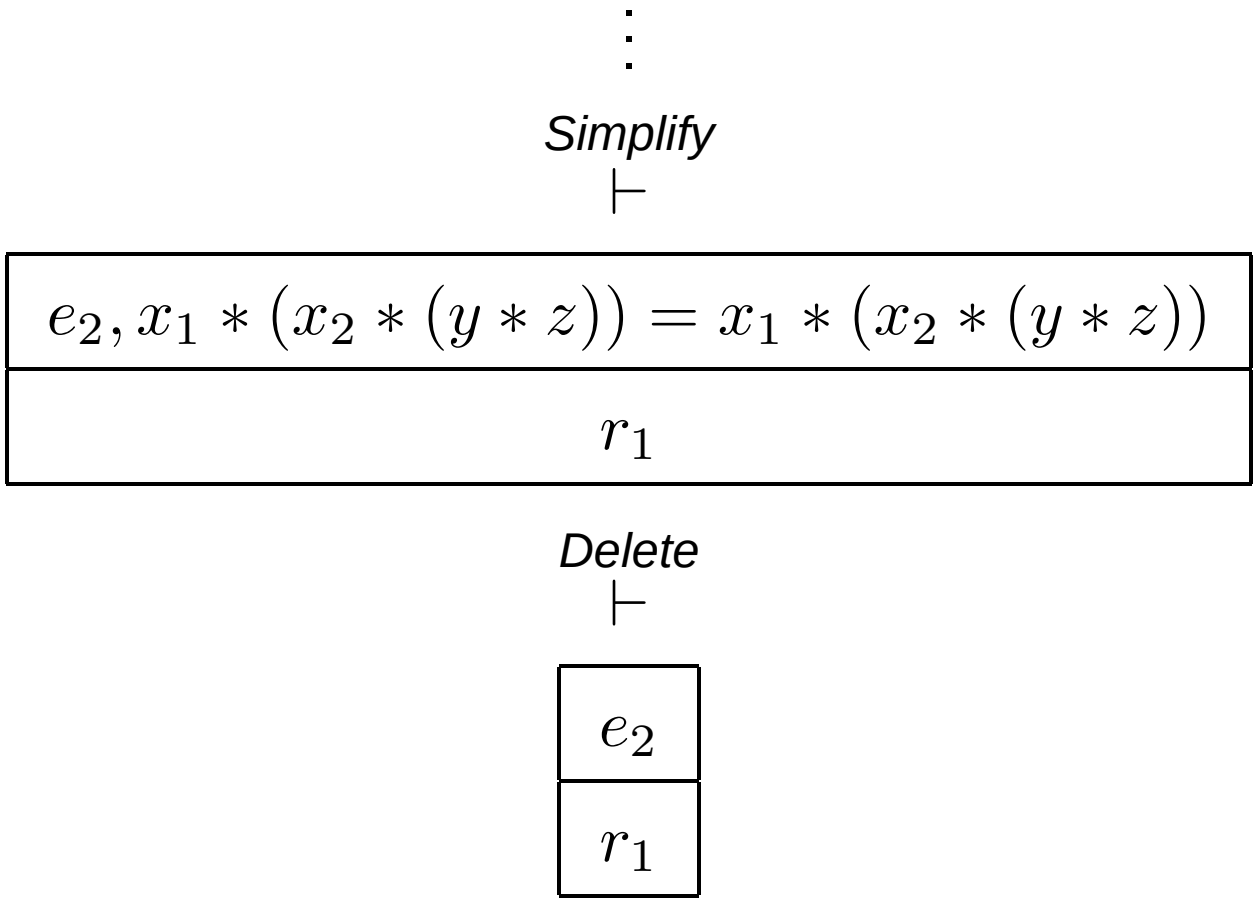
$$R_0 \equiv \emptyset,$$

où

$$\begin{aligned} e_1 &\equiv (x * y) * z &&= x * (y * z) \\ e_2 &\equiv f(x) * f(y) &&= f(x * y) \end{aligned}$$


e_1, e_2
Orient
$\vdash$
e_2
$(x * y) * z \longrightarrow x * (y * z)$
Deduce
$\vdash$
$e_2, (x_1 * (x_2 * y)) * z = (x_1 * x_2) * (y * z)$
r_1
Simplify
$\vdash$

$e_2, x_1 * ((x_2 * y) * z) = (x_1 * x_2) * (y * z)$
r_1


Simplify
$e_2, x_1 * (x_2 * (y * z)) = x_1 * (x_2 * (y * z))$
r_1
Delete
e_2
r_1

Orient

$\vdash$

$r_1, f(x) * f(y) \rightarrow f(x * y)$

Deduce

$\vdash$

$f(x) * (f(y) * z) = f(x * y) * z$
r_1, r_2

Orient

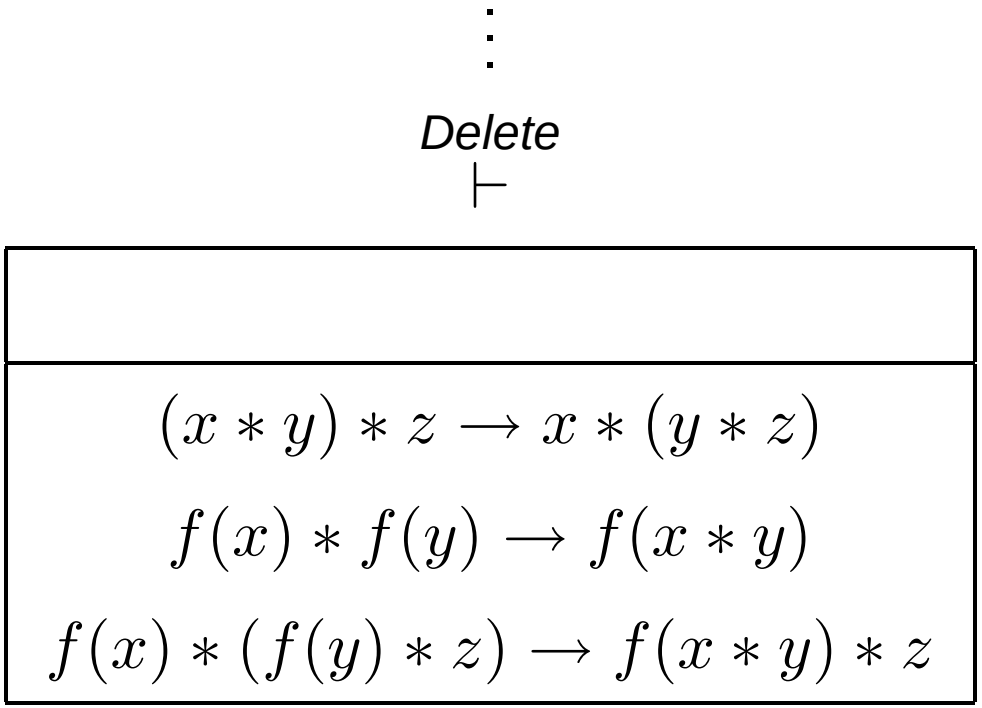
$\vdash$

$r_1, r_2, f(x) * (f(y) * z) \rightarrow f(x * y) * z$

Deduce

$\vdash$

$f(x) * f(y * z) = (f(x) * f(y)) * f(z)$
r_1, r_2, r_3


Delete
$\vdash$
$(x * y) * z \longrightarrow x * (y * z)$
$f(x) * f(y) \longrightarrow f(x * y)$
$f(x) * (f(y) * z) \longrightarrow f(x * y) * z$

Qu'est-ce que la COMPLÉTION ?

des règles de transitions :

- *Orient*,
- *Simplify*,...

un contrôle équitable

une boîte à outils avec

- l'unifi cation,
- le fi ltrage,
- la réécriture,...

+

une structure de donneés ici (R, E) ou (E_raw, E_norm, S, T, C, N, A).

Les ensembles R_j et E_j

Une complétion peut-être infinie.

Au cours de son déroulement, la complétion peut créer une infinité d'ensembles R_j et E_j.

On souhaite

- qu'une règle reste indéfiniment dans les R_j seulement si elle "doit" y rester.
- qu'une identité ne reste pas indéfiniment dans les E_j.

Les ensembles R_j et E_j

Une complétion peut-être infinie.

Au cours de son déroulement, la complétion peut créer une infinité d'ensembles R_j et E_j.

On souhaite

- qu'une règle reste indéfiniment dans les R_j seulement si elle "doit" y rester.
- qu'une identité ne reste pas indéfiniment dans les E_j.

Cela a une conséquence sur le comportement de la complétion, meme si elle finit.

Règles persistantes

Une règle r est persistante si elle appartient indéfiniment aux R_j après avoir été créée, c-à-d qu'il existe i tel que

$$r \in \bigcap_{j \geq i} R_j .$$

L'ensemble de règles limite ou ensemble de règles persistantes est

$$R_\omega = \bigcup_{i \geq 0} \bigcap_{j \geq i} R_j$$

Identités persistantes

Une identité e est persistante si e appartient indéfiniment à l'ensemble E_j une fois qu'elle a été introduite, c-à-d qu'il existe un j tel que $e \in \bigcap_{j \geq i} E_j$.

$$E_\omega = \bigcup_{i \geq 0} \bigcap_{j \geq i} E_j$$

est l'ensemble des identités persistantes.

Paires crititiques persistantes

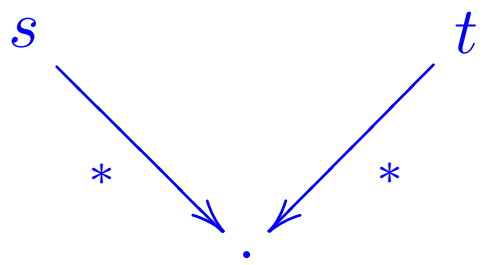
Une paire critique pc est persistante

si pc est une paire critique de R_ω.

$$PC_\omega = pc(R_\omega)$$

Preuve par réécriture

Une preuve par réécriture est une preuve de la forme:



Équité du contrôle

Soit R_E le système de réécriture canonique, (c-à-d confluent, noethérien et interréduit) associé à E et $>$.

S'il existe, R_E est unique.

La complétion est équitable si pour toute preuve par réécriture dans R_E, il existe un i tel que cette preuve est une preuve dans R_i.

Autrement dit, le contrôle est équitable si et seulement si

$$R_E = R_\omega$$

Une condition suffisante d’équité

Un contrôle est équitable si

- Il n’existe pas d’identités persistantes,
- Chaque paire critique persistante est tôt ou tard prise en considération,
- Chaque règle persistante est réduite.

en d’autres termes

$$
\begin{array}{|ccc|}
\hline
E_\omega & = & \emptyset \\
pc(R_\omega) & \subseteq & \bigcup_{i\geq 0} E_i \\
R_\omega & \text{est réduit} & \\
\hline
\end{array}
$$

Un contrexemple

$$
\begin{aligned}
f(g(h(x))) &\rightarrow g(x) \\
g(h(g(x))) &\rightarrow g(g(h(x))) \\
f(g(g(x))) &\rightarrow f(g(x))
\end{aligned}
$$

Si l'on l'on ne prend jamais en compte la troisième règle,

on obtient les règles $f(g^{n+1}(h(x))) \rightarrow g^{n+1}(x)$

Sinon on obtient le système:

$$\begin{aligned} f(g(h(x))) &\rightarrow g(x) \\ g(h(g(x))) &\rightarrow h(x) \\ g(g(x)) &\rightarrow g(x) \end{aligned}$$

puis

$$\begin{aligned} f(h(x)) &\rightarrow g(x) \\ h(g(x)) &\rightarrow h(x) \\ g(h(x)) &\rightarrow h(x) \\ g(g(x)) &\rightarrow g(x) \\ f(g(x)) &\rightarrow g(x) \end{aligned}$$

La construction d'une bonne complétion

- Puisque les règles de transition sont des bons outils de preuves, pourquoi ne seraient-elles pas de bons outils d'implantation ?
- Une bonne structure de données est cruciale
- La complétion est comme une machine avec la structure de données comme états et les règles de transition comme transitions.
- Le contrôle ou la stratégie nous dit comment invoquer les règles,
- La boîte à outils contient les composants de la procédure, ils peuvent être réutilisés

La construction d'une bonne complétion (suite)

- L'efficacité est obtenue par des optimisations de haut niveau,
- La lisibilité rend le programme facile à maîtriser,
- Les optimisations de bas niveau sont obtenues au niveau de la boîte à outil,
- Quand on aura obtenu un programme raisonnablement efficace, le programme CAML pourra servir de documentation pour une implantation rapide.