The Knuth-Bendix Completion Algorithm

William Hua

November 26, 2007

What's a word problem?

What's a word problem?

$\alpha \quad \beta$

What's a word problem?

$$\alpha \stackrel{?}{\equiv} \beta$$

Under a set of identities of the form $\alpha_k \equiv \beta_k$.

What's a word problem?

$$\alpha \stackrel{?}{\equiv} \beta$$

Under a set of identities of the form $\alpha_k \equiv \beta_k$.

Not easy to solve generally.

What's a word?

What's a word?

- Same thing as a term

What's a word?

- Same thing as a term

- Variables: $v_1, v_2, v_3, \ldots$
- Operators: $f_1, \ldots, f_N$
- f_k has degree d_k

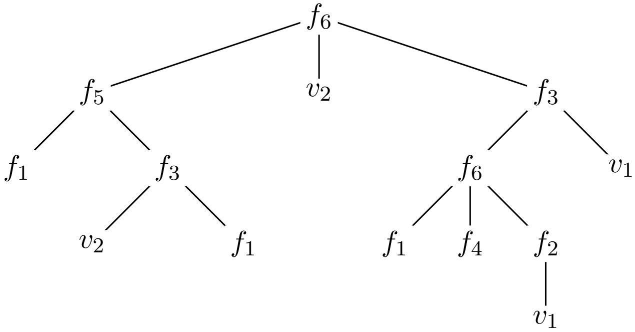
What's a word?

- Same thing as a term

- Variables: $v_1, v_2, v_3, \ldots$
- Operators: $f_1, \ldots, f_N$
- f_k has degree d_k

$$
\begin{aligned}
W &\rightarrow v_k \\
W &\rightarrow f_k \underbrace{W \ldots W}_{d_k}
\end{aligned}
$$

Tree structure



$$f_6 f_5 f_1 f_3 v_2 f_1 v_2 f_3 f_6 f_1 f_4 f_2 v_1 v_1$$

Ordering on words

1. Can find a well-ordering for pure words
2. Can't do this in general for words with variables

Ordering on words

1. Can find a well-ordering for pure words
2. Can't do this in general for words with variables

For an identity $\alpha_k \equiv \beta_k$, assuming $\alpha_k > \beta_k$, we have the *reduction* $\alpha_k \to \beta_k$.

Completeness

Definition

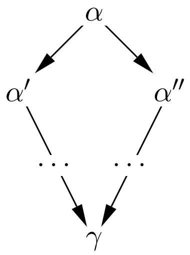
A set of reductions is *complete* if for any irreducible words $\alpha \neq \beta$, we have $\alpha \not\equiv \beta$.

Completeness

Definition

A set of reductions is *complete* if for any irreducible words $\alpha \neq \beta$, we have $\alpha \not\equiv \beta$.

Complete iff the *lattice condition* holds:



Superpositions

$$\sigma(\lambda_1, \mu, \lambda_2)$$

- λ_1 and λ_2 are words
- μ is a subword of λ_2
- λ_1 “looks like” μ

Superpositions

$$\sigma(\lambda_1, \mu, \lambda_2)$$

- λ_1 and λ_2 are words
- μ is a subword of λ_2
- λ_1 "looks like" μ

- Replace the μ in λ_2 with λ_1 to get $\sigma(\lambda_1, \mu, \lambda_2)$
- $\sigma(\lambda_1, \mu, \lambda_2)$ must "look like" λ_2

Let's try it. . .

$$e \cdot a \rightarrow a \quad (1)$$

$$a^{-} \cdot a \rightarrow e \quad (2)$$

$$(a \cdot b) \cdot c \rightarrow a \cdot (b \cdot c) \quad (3)$$

Let's try it. . .

$$e \cdot a \rightarrow a \tag{1}$$

$$a^{-} \cdot a \rightarrow e \tag{2}$$

$$(a \cdot b) \cdot c \rightarrow a \cdot (b \cdot c) \tag{3}$$

$$a^{-} \cdot (a \cdot b) \rightarrow b \tag{4}$$

Let's try it...

$$
\begin{aligned}
e \cdot a &\rightarrow a && (1)\\
a^{-} \cdot a &\rightarrow e && (2)\\
(a \cdot b) \cdot c &\rightarrow a \cdot (b \cdot c) && (3)\\
a^{-} \cdot (a \cdot b) &\rightarrow b && (4)\\
e^{-} \cdot a &\rightarrow a && (5)
\end{aligned}
$$

Let's try it. . .

$$e \cdot a \rightarrow a \quad (1)$$

$$a^{-} \cdot a \rightarrow e \quad (2)$$

$$(a \cdot b) \cdot c \rightarrow a \cdot (b \cdot c) \quad (3)$$

$$a^{-} \cdot (a \cdot b) \rightarrow b \quad (4)$$

$$e^{-} \cdot a \rightarrow a \quad (5)$$

etc.

Until finally. . .

$$
\begin{array}{llcl llcl}
(1) & e \cdot a & \rightarrow & a & (9) & e^{-} & \rightarrow & e \\
(2) & a^{-} \cdot a & \rightarrow & e & (10) & a^{--} & \rightarrow & a \\
(3) & (a \cdot b) \cdot c & \rightarrow & a \cdot (b \cdot c) & (11) & a \cdot a^{-} & \rightarrow & e \\
(4) & a^{-} \cdot (a \cdot b) & \rightarrow & b & (13) & a \cdot (a^{-} \cdot b) & \rightarrow & b \\
(8) & a \cdot e & \rightarrow & a & (20) & (a \cdot b)^{-} & \rightarrow & b^{-} \cdot a^{-}
\end{array}
$$