let $(R, <)$ be an <u>rws</u> on $X^*$

If $R$ is confluent $\Rightarrow$ solving the word problem on $X^*/_R$ is the same as fining the canonical forms w.r.t $<$ on $X^*$ i.e. rewriting words w.r.t. $R$.

If confluence for $R$ fails $\Rightarrow$ local confluence fails at

$$\begin{tikzcd} & W \arrow[dl, "*"'] \arrow[dr, "*"] & \\ u & & v \end{tikzcd}$$

<u>Proposition</u>: Suppose that local confluence fails at $W$, but doesn't for any proper subword of $W$. Then one of the following holds:

1) $W$ is the <u>lhs</u> for two different rules of $R$.
2) $W$ is the <u>lhs</u> for a rewrule in $R$ and $W$ contains <u>lhs</u> of a different rewrule as a <u>proper subword</u>.

3) $W = ABC$, $A, B, C \in X^*$, nonempty & $AB$, $BC$ are <u>lhs</u>es of rewrules from $R$.

---

<u>Proof</u>: By local confuence failure: $\exists$ words $A_1, P_1, B_1$, $A_2, P_2, B_2$ ($A_i, B_i$ - possibly empty) s.t.

- $W = A_1 P_1 B_1 = A_2 P_2 B_2$
- $P_1 \to Q_1$, $P_2 \to Q_2$ $\in R$
- There is no common word derivable from $U_1 = A_1 Q_1 B_1$ and $U_2 = A_2 Q_2 B_2$.

- If the occurrences of $P_1$ an $P_2$ don't overlap

$$\Rightarrow W = A_1 P_1 C P_2 B_2$$

$$\begin{tikzcd}[column sep=small]
 & A_1 P_1 C P_2 B_2 \arrow[dl] \arrow[dr] & \\
A_1 Q_1 C P_2 B_2 \arrow[dr] & & A_1 P_1 C Q_2 B_2 \arrow[dl] \\
 & A_1 Q_1 C Q_2 B_2 &
\end{tikzcd}$$

$\lightning$.

$$\Rightarrow W = A_1 \overbrace{A\,\underbrace{B}_{P_2}}^{P_1} C B_2, \quad B \neq \varepsilon \text{ and either}$$

$(\cdot)$ $P_1 = AB$, $P_2 = BC$ $(A, C \neq \varepsilon)$, or

$(\cdot\cdot)$ $P_1 = ABC$, $P_2 = B$ $(A, C$ possibly empty$)$.

If $A_1 B_2 \neq \varepsilon \Rightarrow ABC$ is a proper subword $\Rightarrow$ local confluence for $ABC$ doesn't fail.

if $(\cdot)$ holds then

$$\begin{tikzcd}[column sep=small]
 & ABC \arrow[dl] \arrow[dr] & \\
Q_1 C \arrow[dr, "*"] & & A Q_2 \arrow[dl, "*"'] \\
 & Q &
\end{tikzcd}$$

hence

$$\begin{tikzcd}[column sep=small]
 & W = A_1\, ABC\, B_2 \arrow[dl] \arrow[dr] & \\
U_1 = A_1 Q_1 C B_2 \arrow[dr] & & U_2 = A_1 A Q_2 B_2 \arrow[dl] \\
 & A_1 Q B_2 &
\end{tikzcd}$$

$\lightning$.

(Similarly for $(\cdot\cdot)$)

Suppose that $A_1 = B_2 = \varepsilon$ i.e. $W = ABC$

Suppose $(\cdot)$ holds.

$AC = \varepsilon \Rightarrow P_1 = P_2 \Rightarrow$ condition (1)

$AC \neq \varepsilon \Rightarrow P_2$ is a proper subword of $P_1$ $\Rightarrow$ condition (2).

$(\cdot\cdot) \Rightarrow$ condition (3).

$\square$

<u>Example</u>:

$X = \{x, y, z\}$, $R = \{xyz \to \varepsilon,\ yzx \to \varepsilon,\ zxy \to \varepsilon\}$

$$\begin{tikzcd}[column sep=small] & \underbrace{xyz}x \arrow[dl] \arrow[dr] & \\ x & \checkmark & x \end{tikzcd} \qquad \begin{tikzcd}[column sep=small] & \underbrace{yzx}y \arrow[dl] \arrow[dr] & \\ y & \checkmark & y \end{tikzcd} \qquad \begin{tikzcd}[column sep=small] & \underbrace{zxy}z \arrow[dl] \arrow[dr] & \\ z & \checkmark & z \end{tikzcd}$$

$$\begin{tikzcd}[column sep=small] & \underbrace{xyz}xy \arrow[dl] \arrow[dr] & \\ xy & \checkmark & xy \end{tikzcd} \qquad \begin{tikzcd}[column sep=small] & \underbrace{yzx}yz \arrow[dl] \arrow[dr] & \\ yz & \checkmark & yz \end{tikzcd} \qquad \begin{tikzcd}[column sep=small] & \underbrace{zxy}zx \arrow[dl] \arrow[dr] & \\ zx & & zx \end{tikzcd}$$

$\Rightarrow R$ is locally confluent.

$F_2 = \langle a, b, A, B \mid Aa = aA = Bb = Bb = \varepsilon \rangle$

$f : F_2 \to \mathrm{Mon}\langle X \mid R \rangle$

$f(a) = x$, $f(b) = y$, $f(A) = yz$, $f(B) = zx$

$g : \mathrm{Mon}\langle X \mid R \rangle \to F_2$

$g(x) = a$, $g(y) = b$, $g(z) = BA$

$$f(g(x)) = f(a) = x$$

$$f(g(y)) = f(b) = y$$

$$f(g(z)) = f(BA) = f(B)\cdot f(A) = zxyz \xrightarrow[R]{} z.$$

$\Rightarrow$ Also $g\circ f = id$ $\Rightarrow$ $X^*/R \cong F_2.$

---

$X = \{x, y, z\}$

$$R = \left\{ \begin{array}{l} x^2 \to \varepsilon \\ yz \to \varepsilon \\ zy \to \varepsilon \end{array} \right\}$$

$$\begin{tikzcd} & x\cdot x\cdot x \arrow[dl] \arrow[dr] & \\ x & \checkmark & x \end{tikzcd} \qquad \begin{tikzcd} & yzy \arrow[dl] \arrow[dr] & \\ y & & y \end{tikzcd} \qquad \begin{tikzcd} & zyz \arrow[dl] \arrow[dr] & \\ z & & z \end{tikzcd}$$

$R$ is locally confluent.

---

$X = \{a, b\}$, $R = \{abab \to \varepsilon, \ baba \to \varepsilon\}$

$$\begin{tikzcd} & ababa \arrow[dl] \arrow[dr] & \\ a & \checkmark & a \end{tikzcd} \qquad \begin{tikzcd} & ababab \arrow[dl] \arrow[dr] & \\ ab & \checkmark & ab \end{tikzcd} \qquad \begin{tikzcd} & abababa \arrow[dl] \arrow[dr] & \\ aba & \checkmark & aba \end{tikzcd}$$

$$\begin{tikzcd} & babab \arrow[dl] \arrow[dr] & \\ b & \checkmark & b \end{tikzcd} \qquad \begin{tikzcd} & bababa \arrow[dl] \arrow[dr] & \\ ba & \checkmark & ba \end{tikzcd} \qquad \begin{tikzcd} & bababab \arrow[dl] \arrow[dr] & \\ bab & \checkmark & bab \end{tikzcd}$$

$R$ is locally confluent

Example:

$$X = \{a, b\} \qquad R = \{\, a^2 \xrightarrow{1} \varepsilon,\ b^3 \xrightarrow{2} \varepsilon,\ abab \xrightarrow{3} b^2a,\ abba \xrightarrow{4} bab,\ baba \xrightarrow{5} ab^2,\ b^2ab^2 \xrightarrow{6} aba \,\}.$$

$$\begin{aligned}
&aaa \xrightarrow{1} a, && aaa \xrightarrow{1} a && \checkmark\\
&aabab \xrightarrow{1} bab, && aabab \xrightarrow{3} abba \xrightarrow{4} bab && \checkmark\\
&aabba \xrightarrow{1} bba, && aabba \xrightarrow{4} abab \xrightarrow{3} bba && \checkmark\\
&bbbb \xrightarrow{2} b, && bbbb \xrightarrow{2} b && \checkmark\\
&bbbbb \xrightarrow{2} b^2, && bbbbb \xrightarrow{2} b^2 && \checkmark\\
&bbbaba \xrightarrow{2} aba, && bbbaba \xrightarrow{5} bbabb \xrightarrow{6} aba && \checkmark\\
&bbbabb \xrightarrow{2} abb, && bbbabb \xrightarrow{6} baba \xrightarrow{5} abb && \checkmark\\
&ababab \xrightarrow{3} bbaab \xrightarrow{1} bbb \xrightarrow{2} \varepsilon, && ababab \xrightarrow{3} abbba \xrightarrow{2} aa \xrightarrow{1} \varepsilon && \checkmark\\
&ababba \xrightarrow{3} bbaba \xrightarrow{5} babb, && ababba \xrightarrow{4} abbab \xrightarrow{4} babb && \checkmark\\
&ababbabb \xrightarrow{3} bbababb \xrightarrow{5} babbbb \xrightarrow{2} bab, && ababbabb \xrightarrow{6} abaaba \xrightarrow{1} abba \longrightarrow bab && \checkmark\\
&ababbb \xrightarrow{3} bbabb \xrightarrow{6} aba, && ababbb \xrightarrow{2} aba && \checkmark\\
&abababa \xrightarrow{3} bbaaba \xrightarrow{1} bbba \xrightarrow{2} a, && abababa \xrightarrow{5} abaabb \xrightarrow{1} abbb \xrightarrow{2} a && \checkmark\\
&ababbabb \xrightarrow{3} bbababb \xrightarrow{3} bbbbab \xrightarrow{2} bab, && ababbabb \xrightarrow{6} abaaba \xrightarrow{1} abba \xrightarrow{4} bab && \checkmark
\end{aligned}$$

...

$X = \{ a, b, A, B \}$

$$R = \{ aA \to \varepsilon,\ bB \to \varepsilon,\ Aa \to \varepsilon,\ Bb \to \varepsilon,$$
$$ba \to ab,\ \boxed{bA \to Ab,}$$
$$Ba \to aB,\ BA \to AB \}$$

$< \; = \text{lenlex}(a < A < b < B)$

$(R, <)$ is locally confluent.

---

$S = R$, ordered by

$\ll \; = \text{lenlex}(a < b < A < B)$

$$ba \to ab,\ \boxed{Ab \to bA}$$
$$Ba \to aB,\ BA \to AB$$

the only difference!

---

$(S, \ll)$ is not <u>locally</u> <u>confluent</u>!

$$\begin{tikzcd}
 & \underbrace{aA}\overbrace{Ab}^{}\!\!\!\!\!\!\!\!\!\!\!\!\! \arrow[dl] \arrow[dr] & \\
b & & abA
\end{tikzcd}$$

$abA \to b$

$$\begin{tikzcd}
 & \underbrace{aA}bb \arrow[dl] \arrow[dr] & \\
bb & & abAb \arrow[d] \\
 & & abbA
\end{tikzcd}$$

$abbA \to bb$

$\vdots$

$ab^nA \to b^n$

this leads to an infinite sequence of rules!

ALGORITHM: isconfluent

INPUT: R - a rewriting system

OUTPUT: true or false and a witness for confluence failure

---

```
begin
for (P1 → Q1) in rwrules(R)
  for S in suffixes(P1, 1:length(P1))
    for (P2 → Q2) in rwrules(R)
      W = lcp(S, P2)   // longest common prefix
      W = ε & continue
      if length(W) = length(S)  // P2 starts with W
        A = P1[1:length(P1)-length(S)]    // P1 = AS
        B = P2[length(S)+1 : length(P2)]  // P2 = SB
        //   ASB   can be rewritten as
        //   ↙   ↘
        //  Q1B    AQ2
        U = Rewrite(Q1B, R); V = Rewrite(AQ2, R)
        U ≠ V & return false, (ASB, U, V)
      elseif length(W) = length(P2) // P2 is a subword of S
        A = P1[1:length(P1)-length(S)]
        B = P1[length(A)+length(W)+1 : length(P1)]
        //  P1 = A·W·B = A·P2·B  rewrites as
        //        ↙         ↘
        //      Q1          A·Q2
        U = Rewrite(Q1, R); V = Rewrite(A·Q2, R)
        U ≠ V & return false, (P1, U, W)
      end
    end
  end
end
return true; end.
```

## Rewriting strategies

Given rws $(R,<)$ and $W$ - a word to be rewritten. How to pick the order in which we choose rewrules in $R$ to do so?

It could be an optimization problem:

- the result minimizes $<$.
- the result minimizes wl.

Since rewriting is done so often we will almost all pick the first one that fits

**but** we may periodically reorder rew-rules of $R$

→ usually we want to sort them w.r.t. wl of the lhs

---

ALGORITHM: destructive_rewrite.

input: $W$ - word to be rewritten

$R$ - rewriting system

output: $V$ - $U \xrightarrow[R]{*} V$

```
begin
  V = one(W)
  while !isone(W)
    x = popfirst!(W)
    push!(V, x)
    for (P → Q) in rwrules(R)
      if P is a suffix of V
        prepend!(W, Q)
        resize!(V, length(V) - length(P))
        break
      end
    end
  end
  return V ; end
```

we are allowed to break here, as all rules of $R$ have been checked against the suffixes of the current $V$.

## Knuth-Bendix procedure

Given an RWS $(R, <)$ we want to compute $RC(R, <)$ – reduced, confluent rws which generates $\sim$ defined by $(R, <)$.

---

**Algorithm : Knuth_Bendix**

**INPUT** : $(R, <)$ – a finite rws

**OUTPUT** : $RC(R, <)$ – reduced, confluent rws

```
begin
  S = Rewriting system ( )
  for (P → Q) in rwrules(R)
      push!(S, P → Q)
  end
  for R1 in rwrules(S)
      for R2 in rwrules(S)
          resolve_overlaps!(S, R1, R2).
          if R1 = R2
              break
          end
          resolve_overlaps!(S, R2, R1)
      end
  end
  return reduce(S)
end
```

## Algorithm: push!

```
INPUT   : (R, <) - rewriting system
          P → Q  - rwrule

OUTPUT  : (R, <) which contains (P → Q).

begin
    U = rewrite(P, R)
    V = rewrite(Q, R)
    if U ≠ V
        U, V = U > V ? (U, V) : (V, U)
        add U → V to rewriting rules of R.
    end
    return R
end
```

## Algorithm: resolve-overlaps

```
INPUT   : (R, <) - rws
          P1 → Q1  - rwrule
          P2 → Q2  - rwrule

OUTPUT  : (R, <) s.t. all rewrites using the
          rules above are locally confluent

begin
    for S in suffixes(P1, 1:length(P1))
        if isprefix(S, P2)                     //   ASB
            push!(R, Q1B → AQ2)                //   Q1B   AQ2
        elseif occursin(P2, S)
            push!(R, Q1 → AQ2B)                //  P1 = AP2B
        end
    end
    return R
end
```

ALGORITHM: reduce

INPUT: $(R,<)$ - an rws

OUTPUT: $(S,<)$ - a reduced version of $(R,<)$

---

```
Begin
    S = empty(R)
    for (P -> Q) in rewrules(R)
        for P' in proper_subwords(P)
            if is_reducible(P',R)
                break
            end
        end
        push!(S, P -> rewrite(R,Q))
    end
    return S
end
```

---

Proposition:

If $RC(R,<)$ is finite, then Knuth_Bendix terminates and returns it.

Proof: Since $S$ is initially a subsystem, rewrites of $P \to Q$ in $S$ follow also in $R$, so that we didn't change the equivalence relation $\sim$ generated by $R$.

During the for loops this property is preserved.

Suppose that Knuth_Bendix doesn't terminate.

$\Rightarrow$ there's an infinite sequence of rules $\mathcal{U}$ added to $S$.

**Prop:** $\mathcal{U} \cup \{\text{the initial rules}\}$ form a confluent rewriting system.

**Proof:** If $\mathcal{U}$ is not confluent let $W$ be the least ($<$) word for which local confluence fails.

We know that $W = ABC$, where $B \neq \varepsilon$ and either

- $P_j = ABC$, $P_i = B$ and $Q_j \xrightarrow[\not *]{\mathcal{U}} Q \xleftarrow[\not *]{\mathcal{U}} AQ_iC$

•• $P_j = AB$, $P_i = BC$ and $Q_jC \xrightarrow[\not *]{\mathcal{U}} Q \xleftarrow[\not *]{\mathcal{U}} AQ_i$

- When resolve_overlaps$(S, P_j \to Q_j, P_i \to Q_i)$ is called a rule which guarantees the existence of $Q$ would be added to current $S$. since the current $S \subset \mathcal{U}$ $\to$ contradition

(similarly for ••)

$\square$

Let $\mathcal{C}$ - canonical forms for $\sim_{R}$.

Let $\mathcal{L} = \{L \in X^* \setminus \mathcal{C} :$ every proper subword of $L$ is in $\mathcal{C}\}$

By confluence: rewriting $L$ must produce $\bar{L} \in \mathcal{C}$ $\Rightarrow$ $L \to \bar{L}$ belongs to $\mathcal{U}$.

By assumption $RC(R,<)$ is finite $\Rightarrow$ $\mathcal{T} = \{L \to \bar{L}\}$ is finite so we'll see all those rules in finite time.

If $P \to Q$ is added later in the process $\Rightarrow P \notin \mathcal{C}$ and $P$ is irreducible w.r.t. $\mathcal{T}$ $\lightning$ contradict with confluence of $RC(R,<)$.

Example 1: $a < A < b$

1. $aA \to \varepsilon$
2. $Aa \to \varepsilon$
3. $bb \to \varepsilon$
4. $ba \to ab$
5. $abA \to b$
6. $bA \to Ab$

(1,1) nothing

(1,2) $a \leftarrow aAa \to a$ confluent

(2,1) $A \leftarrow AaA \to A$ confluent

(2,2) nothing

(3,1), (1,3), (3,2), (2,3): nothing

(3,3): $b \leftarrow bbb \to b$ confluent

(4,1) $abA \leftarrow baA \to b$ $\Rightarrow$ new rule: 5

(1,4), (4,2), (2,4): nothing

(4,3): nothing

(3,4): $a \leftarrow bba \xrightarrow{1} bab \xrightarrow{4} abb \xrightarrow{3} a$ confluent

(4,4): nothing

(5,1), (1,5) nothing

(5,2) $ab \leftarrow ba \leftarrow abAa \to ab$ confluent

(2,5) $bA \leftarrow AabA \to Ab$ $\Rightarrow$ new rule: 6

(5,3), (3,5) nothing

(5,4) nothing

(4,5) $\varepsilon \xleftarrow{*} abbA \leftarrow babA \to bb \to \varepsilon$ confluent

(5,5) nothing

(6,1), (1,6) nothing

(6,2): $b \xleftarrow{*} Aba \leftarrow bAa \to b$ confluent

(2,6), (6,3) nothing

(3,6) $A \leftarrow bbA \to bAb \to Abb \to A$ confluent

(6,4) (4,6) nothing

(6,5) nothing

(5,6) $b \leftarrow abA \to aAb \to b$ confluent

(6,6) nothing

- LenLex$(a < A < b < B)$
- $R = \{$

1. $aA \to \varepsilon$
2. $Aa \to \varepsilon$
3. $bB \to \varepsilon$
4. $Bb \to \varepsilon$

---

5. $ba \to ab$
6. $abA \to b$
7. $Bab \to a$

$(1,1)$: nothing

$(2,1)$: $A \leftarrow AaA \to A$ confluent

$(1,2)$: $a \leftarrow aAa \to a$ confluent

$(2,2)$: nothing

$(3,1),(1,3)$: nothing

$(3,2),(2,3)$: nothing

$(3,3)$: nothing

$(4,1),(1,4)$: nothing

$(4,2),(2,4)$: nothing

$(4,3)$: $B \leftarrow BbB \to B$ confluent

$(3,4)$: $b \leftarrow bBb \to b$ confluent

$(4,4)$: nothing

$(5,1)$: $abA \leftarrow baA \to b \Rightarrow$ new rule: 6

$(1,5)$: nothing

$(5,2),(2,5)$: nothing

$(5,3),(3,5)$: nothing

$(5,4)$: nothing

$(4,5)$ $a \leftarrow Bba \to Bab \Rightarrow$ new rule: 7

$(5,5)$ nothing

ng

- LenLex$(a < A < b < B)$
- $\mathcal{R} = \{$
  1. $aA \to \varepsilon$
  2. $Aa \to \varepsilon$
  3. $bB \to \varepsilon$
  4. $Bb \to \varepsilon$
  5. $ba \to ab$
  6. $abA \to b$
  7. $Bab \to a$
  8. $bA \to Ab$
  9. $Ba \to aB$
  10. $BAb \to A$

(6,1),(1,6): nothing
(6,2) : $ab \xleftarrow{*} abAa \to ba$ confluent
(2,6) : $bA \leftarrow AabA \to Ab \Rightarrow$ new rule : 8
(6,3), (3,6), (6,4),(4,6) → nothing
(6,5) : nothing
(5,6) : $bb \xleftarrow{*} babA \to bb$ confluent
(6,6) : nothing

(7,1), (1,7), (2,7), (7,2): nothing
(7,3) : $aB \leftarrow BabB \to Ba \Rightarrow$ new rule 9
(3,7) : $ab \leftarrow bBab \xrightarrow{*} ab$ confluent
(7,4), (4,7): nothing
(7,5) : $aa \leftarrow Baba \to Baab \xrightarrow{*} aa$ confluent
(5,7) : nothing
(7,6) : $\varepsilon \xleftarrow{*} BabA \xrightarrow{*} \varepsilon$ confluent
(7,7) : nothing

(8,1), (1,8) : nothing
(8,2) : $b \xleftarrow{*} bAa \to b$ confluent
(2,8) : nothing
(8,3), (3,8) : nothing
(8,4) nothing
(4,8) : $A \leftarrow BbA \to BAb \Rightarrow$ new rule 10
(8,5), (5,8) : nothing

- LenLex($a < A < b < B$)
- $\mathcal{R} = \{$

| | | |
|---|---|---|
| 1 | $aA$ | $\to \varepsilon$ |
| 2 | $Aa$ | $\to \varepsilon$ |
| 3 | $bB$ | $\to \varepsilon$ |
| 4 | $Bb$ | $\to \varepsilon$ |
| 5 | $ba$ | $\to ab$ |
| 6 | $abA$ | $\to b$ |
| 7 | $Bab$ | $\to a$ |
| 8 | $bA$ | $\to Ab$ |
| 9 | $Ba$ | $\to aB$ |
| 10 | $BAb$ | $\to A$ |
| 11 | $aBA$ | $\to B$ |
| 12 | $BA$ | $\to AB$ |

(8,6), (6,8) : nothing

(8,7) : nothing

(7,8) $\varepsilon \xleftarrow{*} BabA \to BaAb \xrightarrow{*} \varepsilon$ confluent

(8,8) : nothing

(9,1): $aBA \leftarrow BaA \to B \Rightarrow$ new rule 11

(1,9), (9,2) (2,9) (9,3): nothing

(3,9) $a \leftarrow bBa \to baB \xrightarrow{*} a$ confluent

(9,4) (4,9), (9,5), (5,9) : nothing

(9,6) $\varepsilon \xleftarrow{*} BabA \to Bb \to \varepsilon$ confluent

(6,9) nothing

(9,7) $a \xleftarrow{*} Bab \to a$ conflict

(7,9), (9,8), (8,9), (9,9) nothing

(10,1), (1,10), (10,2), (2,10) : nothing

(10, 3): $AB \leftarrow BAbB \to BA \Rightarrow$ new rule: 12

⋮ ⋮

This is the last rule we add; following critical pairs will lead to confluent rewrites.

The reduced rws consists of rules:

1, 2, 3, 4, 5, 8, 9, 12.

Second version:

Keep the set of rules always reduced.

Whenever we add a new rule - scan all of the others to determine those which become simpler / redundant. ⇒ push them to a stack ⇒ operate until stack is empty

ALGORITHM: append!

INPUT : • $(R,<)$ - reduced rws
• stack - a list of rules to be added

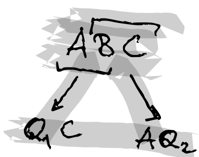
OUTPUT : • $(R,<)$ - reduced rws

```
begin
  while !isempty(stack)
      P → Q = pop!(stack)
      A = rewrite(P, R); B = rewrite(Q, R)
      if A ≠ B
         A,B = A > B ? (A,B) : (B,A)
         add A → B as a rule to R
         for P → Q in active_rules(R)
             (P → Q) = (A → B) && continue
             if occursin(A, P)      // rule is reducible
                push!(stack, P → Q)
                deactivate!(R, P → Q)
             elseif occursin(A, Q)
                rewrite!(Q, A → B)  } in place
                rewrite!(Q, R)      } modifications
             end
         end
      end
  return R
end
```

ALGORITHM: resolve_overlaps!

INPUT : • (R, <) – reduced rws
• ($P_1 \to Q_1$) – rwrule
• ($P_2 \to Q_2$) – rwrule
• stack – a stack of rules

OUTPUT : (R, <) – rws where all critical pairs from $P_1$ and $P_2$ are resolved




```
begin
   m = min(length(P1), length(P2))
   while isactive(P1 → Q1) && isactive(P2 → Q2)
      for B in suffixes(P1, 1:m-1)
         if isprefix(B, P2)
            A = P1[1:length(P1) - length(B)]
            B = P2[length(B)+1 : length(P2)]
            push!(stack, AQ2 → Q1C)   # any ordering is fine
            append!(R, stack)
         end
      end
   end
   return R
end
```

ALGORITHM : Knuth-Bendix_always_reduced

INPUT : $(\mathcal{R}, <)$ - rws

OUTPUT : $RC(\mathcal{R}, <)$ - the unique, reduced, confluent rws.

```
begin
    stack = ∅
    for r in rules(R)
        push!(stack, r)
    end
    S = empty(R)
    append!(S, stack)

    for r1 in active_rules(S)
        for r2 in active_rules(S)
            isactive(r1) || break
            resolve_overlaps!(S, r1, r2, stack)
            r1 = r2 && break
            isactive(r2) || continue
            isactive(r1) || break
            resolve_overlaps!(S, r2, r1, stack)
        end
    end
    delete inactive rules from S
    return S
end
```

Example:

$a^2 \to \varepsilon$

$b^3 \to \varepsilon$

$(ab)^7 \to \varepsilon$

$(abab^2)^8 \to \varepsilon$

hard

Hurwitz groups, i.e quotients of (2,3,7) triangle groups.

with

1, 2, 3, 5 – collapses

4 – 40 rules

6 – 119 rules

7 – 147 rules

8 – ???

9 – ???