1 Explanation Generation

Expliciet zeggen: C a cnf. U a set of vars

INPUT

Algorithm 1: EXPLAINCSP($\mathcal{C}, U, f$) I

// Hyp: f

$E \leftarrow \langle\rangle$

$I_{end} \leftarrow$ OPTIMALPROPAGATE($\mathcal{C}, U$) // assignment on variables of U

C ∪ I

bart: What's a better way to get the initial interpr.?

$I \leftarrow \{i \in I_{end} | f(-i) = inf\}$ // contains the indicator literals and the initial assignment

BAH

while $I \neq I_{end}$ **do**

  $X \leftarrow$ BESTSTEP($\mathcal{C}, f, I_{end}, I$)

  $I_{best} \leftarrow I \cap X$

  $N_{best} \leftarrow \{\cancel{I_{end} \setminus I}\} \cap$ OPTIMALPROPAGATE($C \cup I_{best}, U$) $\setminus I$

  add $\{I_{best} \implies N_{best}\}$ to E

  $I \leftarrow I \cup N_{best}$

end

return E

gebruik P voor literal

Algorithm 2: BESTSTEP–C-OUS($\mathcal{C}, f, I_{end}, I$)

$A \leftarrow I \cup (\overline{I_{end}} \setminus \bar{I})$ // Optimal US is subset of A

set p such that exactly one of $\overline{I_{end}}$ in the hitting set

return C-OUS($\mathcal{C}, f, p, A$)

p is the constraint $\#\{\overline{I_{end}}\} = 1$

$\sum_{l \in \overline{I_{end}}} l = 1$

Algorithm 3: C-OUS($\mathcal{C}, f, p, A$)

// Hyp: $\mathcal{C} \cup A$ is unsatisfiable

$\mathcal{H} \leftarrow \emptyset$

while *true* **do**

nodig ?

  $A' \leftarrow$ CONDOPTHITTINGSET($f, p, A, \mathcal{H}$)

  if $\neg$SAT($\mathcal{C} \cup A'$) **then**

    return A'

  end

  $A'' \leftarrow$ GROW(A', A)

HOE WERKT DIT ? C nodig ? p nodig ? f nodig ?

  $\mathcal{H} \leftarrow \mathcal{H} \cup \{A \setminus A''\}$ // We can reuse the H across diff call to alg 1 was $H \cup \{F \setminus F''\}$

end

Algorithm 4: OPTIMALPROPAGATE($\mathcal{C}, \mathcal{U}$)

$sat?, \mu \leftarrow$ SAT($\mathcal{C}$)

$\mu \leftarrow \mu \cap \mathcal{U}$

DOES NOT PARSE: μ = set literals ; U = set vars

Should I add the blocking variables b_i to the newly added clause ?

!

while *true* **do**

  $\mathcal{C} \leftarrow \mathcal{C} \wedge (\bigvee_{1 \leq i \leq |\mu|} \neg x_i)$

  or $\mathcal{C} \leftarrow \mathcal{C} \wedge (\bigvee_{1 \leq i \leq |\mu|} \neg x_i)$

  or $\mathcal{C} \leftarrow \mathcal{C} \wedge (\neg x_1 \ldots \vee \neg x_{|\mu|})$

  or $\mathcal{C} \leftarrow \mathcal{C} \wedge (\neg b_i \vee \bigvee_{1 \leq i \leq |\mu|} \neg x_i)$

  or $\mathcal{C} \leftarrow \mathcal{C} \wedge (\neg b_i \vee \neg x_1 \ldots \vee \neg x_{|\mu|})$

?

wat zegt dit? doel? wil je je solver herbruiken ?

indien ja: $\boxed{1}$ blocking var

prev

- SAT call met alle vars als assumptie

-

  $sat?, \mu' \leftarrow$ SAT($\mathcal{C}$)

  if $\neg sat?$ **then**

    return μ

  end

  $\mu \leftarrow \mu \cap \mu' \cap \mathcal{U}$

end

$\bigvee_{x \in \mu} \neg x$

gebruik geen index indien niet nodig

2 OUS Algorithm

Algorithm 5: $\textsc{bestStep}$–OUS($\mathcal{C}, f, I, I_{end}$)

$X_{best} \leftarrow \{\mathcal{C} \land I \land \overline{I_{end}}\}$
for $l \in \{I_{end} \setminus I\}$ **do**
  $X \leftarrow \text{OUS}(\mathcal{C} \land I \land \neg l,\ f)$
  if $f(X) < f(X_{best})$ **then**
    $X_{best} \leftarrow X$
  end
end
return X_{best}

Algorithm 6: OUS-$\textsc{Inc}(\mathcal{F}, f)$

SSOfF $\leftarrow \emptyset$
for $S \in$ **SSs** **do**
  $S_{\mathcal{F}} \leftarrow S \cap \mathcal{F}$
  if $\neg\exists S' \in$ **SSOfF** : $S_{\mathcal{F}} \subseteq S'$ **then**
    $S_{\mathcal{F}} \leftarrow \textsc{Grow}(S_{\mathcal{F}}, \mathcal{F})$
    $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{F} \setminus S_{\mathcal{F}}\}$
    SSOfF $\leftarrow$ **SSOfF** $\cup \{S_{\mathcal{F}}\}$
  end
end
while *true* **do**
  $\mathcal{F}' \leftarrow \textsc{OptHittingSet}(\mathcal{H}, f)$
  if $\neg\textsc{sat}(\mathcal{F}')$ **then**
    return $\mathcal{F}'$
  end
  $\mathcal{F}'' \leftarrow \textsc{Grow}(\mathcal{F}', \mathcal{F})$
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{F} \setminus \mathcal{F}''\}$
  SSs $\leftarrow$ **SSs** $\cup \{(\mathcal{F}'', M)\}$
end

Algorithm 7: Postponing hitting set optimization for OUS (to be inserted before of)

while *true* **do**
  while $|\mathcal{H}| > 0$ **do**
    $\mathcal{F}' \leftarrow \mathcal{F}' + min_f$ element of last MCS in $\mathcal{H}$;
    if $\neg\textsc{sat}(\mathcal{F}')$ **then**
      break
    end
    $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{F} \setminus \textsc{Grow}(\mathcal{F}', \mathcal{F})\}$;
  end
  $\mathcal{F}' \leftarrow \textsc{GreedyHittingSet}(\mathcal{H}, f)$;
  if $\neg\textsc{sat}(\mathcal{F}')$ **then**
    break
  end
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{\mathcal{F} \setminus \textsc{Grow}(\mathcal{F}', \mathcal{F})\}$;
end