Theorem Proving Using Constraint Satisfaction

David E. Narváez
den9562@rit.edu

Topics in Advanced Algorithms, Spring 2021

- Constraint Satisfaction Techniques try to find models that satisfy a set of constraints.
- Constraint Satisfaction Problems (CSPs) can be of several types, each type called a *paradigm*.

Encoding Combinatorial Problems via CSPs

Overall Strategy

Phases:

- Find a suitable constraint satisfaction paradigm.
- Devise a formula that represents the combinatorial problem in the selected paradigm.
- Encode the problem as a formula.
- Use a solver for the selected paradigm.
  - If satisfiable: decode the satisfying assignment.
  - If unsatisfiable: provide a proof of unsatisfiability.

Encoding Combinatorial Problems as a CSP

But Why?

- High availability of solvers, developed independently from problem encodings.
- Several success stories in the last few years.
- Results are independently *verifiable*.

Encoding Combinatorial Problems as SAT

Running Examples

We will be proving theorems in *Ramsey theory*. Here are some definitions:

- K_n is the complete graph on n vertices.
- $J_n = K_n - e$ is the complete graph minus one edge.

Encoding Combinatorial Problems as SAT

Running Examples

We will be proving theorems in *Ramsey theory*. Here are some definitions:

- K_n is the complete graph on n vertices.
- $J_n = K_n - e$ is the complete graph minus one edge.
- The notation $G \to (H_1, H_2)$ means that:
  In any coloring of the edges of G with two colors,
  there will be an H_1 in the 1st color or an H_2 in the 2nd color.

Encoding Combinatorial Problems as SAT

Running Examples

We will be proving theorems in *Ramsey theory*. Here are some definitions:

- K_n is the complete graph on n vertices.
- $J_n = K_n - e$ is the complete graph minus one edge.
- The notation $G \to (H_1, H_2)$ means that:
  In any coloring of the edges of G with two colors,
  there will be an H_1 in the 1st color or an H_2 in the 2nd color.
- The Ramsey number $R(H_1, H_2)$ is the order of the smallest K_n such that $K_n \to (H_1, H_2)$.

Encoding Combinatorial Problems as SAT

Running Examples

We will be proving theorems in *Ramsey theory*. Here are some definitions:

- K_n is the complete graph on n vertices.
- $J_n = K_n - e$ is the complete graph minus one edge.
- The notation $G \to (H_1, H_2)$ means that:
  In any coloring of the edges of G with two colors,
  there will be an H_1 in the 1st color or an H_2 in the 2nd color.
- The Ramsey number $R(H_1, H_2)$ is the order of the smallest K_n such that $K_n \to (H_1, H_2)$.

Encoding Combinatorial Problems as SAT

Running Example

The Ramsey number $R(K_3, K_3)$ is order of the smallest K_n such that $K_n \to (K_3, K_3)$.

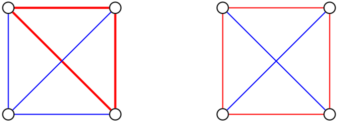

Figure: Two ways to color the edges of K_4.

The Ramsey number $R(K_3, K_3)$ is order of the smallest K_n such that $K_n \to (K_3, K_3)$.

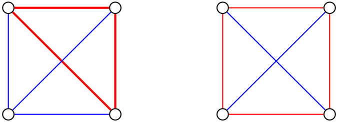

Figure: Two ways to color the edges of K_4.

Note there are no triangles in the coloring to the right, so $R(K_3, K_3) > 4$.

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, K_3) = 6$

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, K_3) = 6$

We need to prove two things:

- There is a coloring of K_5 that has no triangles of the same color.
- There is not a coloring of K_6 that has no triangles of the same color.

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, K_3) = 6$

Phases:

- Find a suitable constraint satisfaction paradigm.
- Devise a formula that represents the combinatorial problem in the selected paradigm.
- Encode the problem as a formula.
- Use a solver for the selected paradigm.
  - If satisfiable: decode the satisfying assignment.
  - If unsatisfiable: provide a proof of unsatisfiability.

Boolean Satisfiability (SAT)

Boolean formulas in *conjunctive normal form (CNF)*, i.e., restricted to conjunctions ($\wedge$) of disjunctions ($\vee$). Ex:

$$(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y})$$

Boolean Satisfiability (SAT)

Boolean formulas in *conjunctive normal form (CNF)*, i.e., restricted to conjunctions ($\wedge$) of disjunctions ($\vee$). Ex:

$$(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y})$$

- The variables are x, y and z.
- The literals are the variables and their negations, e.g., $\overline{z}$.
- Each disjunction is called a clause.
  - $x \vee y \vee \overline{z}$
  - $\overline{x} \vee \overline{y}$
- A formula is satisfiable if there is an assignment of the variables such that the formula evaluates to true.
  - It is unsatisfiable otherwise.

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, K_3) = 6$

Phases:

- Find a suitable constraint satisfaction paradigm.
- Devise a formula that represents the combinatorial problem in the selected paradigm.
- Encode the problem as a formula.
- Use a solver for the selected paradigm.
  - If satisfiable: decode the satisfying assignment.
  - If unsatisfiable: provide a proof of unsatisfiability.

Encoding Combinatorial Problems as SAT

Running Example

Assign a Boolean variable to each edge:

- $x_{i,j}$ represents the color of the edge between vertices i and j.
- If the value of $x_{i,j}$ is *true*, we color the edge between i and j red.
- If the value of $x_{i,j}$ is *false*, we color the edge between i and j blue.

Encoding Combinatorial Problems as SAT

Running Example

Assign a Boolean variable to each edge:

- $x_{i,j}$ represents the color of the edge between vertices i and j.
- If the value of $x_{i,j}$ is *true*, we color the edge between i and j red.
- If the value of $x_{i,j}$ is *false*, we color the edge between i and j blue.

For every triple of vertices i, j, k, the edges between them are a potential triangle, so:

- At least one of the edges has to be red $\Rightarrow$ at least one of $x_{i,j}$, $x_{j,k}$, $x_{i,k}$ has to be *true*.
- At least one of the edges has to be blue $\Rightarrow$ at least one of $x_{i,j}$, $x_{j,k}$, $x_{i,k}$ has to be *false*.

Encoding Combinatorial Problems as SAT

Running Example

Assign a Boolean variable to each edge:

- $x_{i,j}$ represents the color of the edge between vertices i and j.
- If the value of $x_{i,j}$ is *true*, we color the edge between i and j red.
- If the value of $x_{i,j}$ is *false*, we color the edge between i and j blue.

For every triple of vertices i, j, k, the edges between them are a potential triangle, so:

- At least one of the edges has to be red $\Rightarrow$ at least one of $x_{i,j}$, $x_{j,k}$, $x_{i,k}$ has to be *true*.
  $(x_{i,j} \vee x_{j,k} \vee x_{i,k})$
- At least one of the edges has to be blue $\Rightarrow$ at least one of $x_{i,j}$, $x_{j,k}$, $x_{i,k}$ has to be *false*.
  $(\overline{x_{i,j}} \vee \overline{x_{j,k}} \vee \overline{x_{i,k}})$

Encoding Combinatorial Problems as SAT

Running Example

The formula we need is:

$$F_N = \forall\ i < j < k < N.\ (x_{i,j} \vee x_{j,k} \vee x_{i,k}) \wedge (\overline{x_{i,j}} \vee \overline{x_{j,k}} \vee \overline{x_{i,k}})$$

F_N is satisfiable if and only if K_N can be colored in a way that avoids triangles of the same color.

Encoding Combinatorial Problems as SAT

Running Example

The formula we need is:

$$F_N = \forall\ i < j < k < N.\ (x_{i,j} \vee x_{j,k} \vee x_{i,k}) \wedge (\overline{x_{i,j}} \vee \overline{x_{j,k}} \vee \overline{x_{i,k}})$$

F_N is satisfiable if and only if K_N can be colored in a way that avoids triangles of the same color.

We need to prove two things:

- There is a coloring of K_5 that has no triangles of the same color $\Rightarrow F_5$ is satisfiable.
- There is not a coloring of K_6 that has no triangles of the same color $\Rightarrow F_6$ is unsatisfiable.

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, K_3) = 6$

Phases:

- Find a suitable constraint satisfaction paradigm.
- Devise a formula that represents the combinatorial problem in the selected paradigm.
- Encode the problem as a formula.
- Use a solver for the selected paradigm.
  - If satisfiable: decode the satisfying assignment.
  - If unsatisfiable: provide a proof of unsatisfiability.

Enter Python 3 and `itertools`.

```
  $ python3
Python 3.7.8 (default, Aug 26 2020, 17:06:51)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.

>>> import itertools
>>> for c in itertools.combinations(['a', 'b', 'c', 'd'], 2):
...   print(c)
...
('a', 'b')
('a', 'c')
('a', 'd')
('b', 'c')
('b', 'd')
('c', 'd')
```

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, K_3) = 6$

Phases:

- Devise a Boolean formula that represents the combinatorial problem.
  - Convert the formula to CNF.
- Encode the problem as a CNF formula.
- Use a SAT solver.
  - If satisfiable: decode the satisfying assignment.
  - If unsatisfiable: provide a proof of unsatisfiability.

Encoding Combinatorial Problems as SAT

Running Example

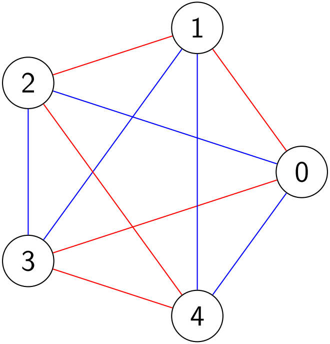

Figure: The satisfying assignment for K_5.

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, J_4) = 7$

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, J_4) = 7$

We need to prove two things:

- There is a coloring of K_6 that has no K_3 in the first color and no J_4 in the second color.
- There is not a coloring of K_7 that has no K_3 in the first color and no J_4 in the second color.

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, J_4) = 7$

Phases:

- Find a suitable constraint satisfaction paradigm.
- Devise a formula that represents the combinatorial problem in the selected paradigm.
- Encode the problem as a formula.
- Use a solver for the selected paradigm.
  - If satisfiable: decode the satisfying assignment.
  - If unsatisfiable: provide a proof of unsatisfiability.

Encoding Combinatorial Problems as SAT

Running Example

Take a closer look at the Boolean constraints for J_4: at least 2 of the $\binom{4}{2}$ edges of K_4 are **not** blue.

Take a closer look at the Boolean constraints for J_4: at least 2 of the $\binom{4}{2}$ edges of K_4 are **not** blue.

$$\begin{aligned}\forall i_1 < i_2 < i_3 < i_4.\ &(\overline{x_{i_1,i_2}} \wedge \overline{x_{i_2,i_3}}) \vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_2,i_4}}) \vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_1,i_3}}) \\ &\vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_1,i_4}}) \vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_2,i_3}}) \vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_2,i_4}}) \\ &\vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_2,i_4}}) \vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_2,i_4}}) \vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_1,i_4}}) \\ &\vdots\end{aligned}$$

Encoding Combinatorial Problems as SAT

Running Example

Take a closer look at the Boolean constraints for J_4: at least 2 of the $\binom{4}{2}$ edges of K_4 are **not** blue.

$$\forall i_1 < i_2 < i_3 < i_4.\ (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_2,i_3}}) \vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_2,i_4}}) \vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_1,i_3}})$$
$$\vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_1,i_4}}) \vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_2,i_3}}) \vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_2,i_4}})$$
$$\vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_2,i_4}}) \vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_2,i_4}}) \vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_1,i_4}})$$
$$\vdots$$

- Disjunction of $\binom{6}{2}$ conjunctions, so not in CNF.

Encoding Combinatorial Problems as SAT

Running Example

Take a closer look at the Boolean constraints for J_4: at least 2 of the $\binom{4}{2}$ edges of K_4 are **not** blue.

$$\begin{aligned}\forall i_1 < i_2 < i_3 < i_4.\ &(\overline{x_{i_1,i_2}} \wedge \overline{x_{i_2,i_3}}) \vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_2,i_4}}) \vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_1,i_3}})\\ &\vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_1,i_4}}) \vee (\overline{x_{i_1,i_2}} \wedge \overline{x_{i_2,i_3}}) \vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_2,i_4}})\\ &\vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_2,i_4}}) \vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_2,i_4}}) \vee (\overline{x_{i_2,i_3}} \wedge \overline{x_{i_1,i_4}})\\ &\vdots\end{aligned}$$

- Disjunction of $\binom{6}{2}$ conjunctions, so not in CNF.
- Fortunately, the presence of J_4 can be entirely characterized by the number of edges present.

Pseudo-Boolean Satisfiability (PB)

Collection of pseudo-Boolean constraints of the form $\sum c_i * x_i \geq l_i$, where x_i is either 0 or 1 and c_i is an integer. E.g.:

$$3x + 2y + -1z \geq 5$$

Pseudo-Boolean Satisfiability (PB)

Collection of pseudo-Boolean constraints of the form $\sum c_i * x_i \geq l_i$, where x_i is either 0 or 1 and c_i is an integer. E.g.:

$$3x + 2y + -1z \geq 5$$

Note that CNF clauses

$$(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y})$$

can be interpreted as:

$$\begin{aligned} 1x + 1y + (1 - z) &\geq 1 \\ (1 - x) + (1 - y) &\geq 1 \end{aligned}$$

Pseudo-Boolean Satisfiability (PB)

Collection of pseudo-Boolean constraints of the form $\sum c_i * x_i \geq l_i$, where x_i is either 0 or 1 and c_i is an integer. E.g.:

$$3x + 2y + -1z \geq 5$$

Note that CNF clauses

$$(x \vee y \vee \overline{z}) \wedge (\overline{x} \vee \overline{y})$$

can be interpreted as:

$$\begin{aligned} 1x + 1y + (1 - z) &\geq 1 \\ (1 - x) + (1 - y) &\geq 1 \end{aligned}$$

- A PB formula is satisfiable if there is an assignment of the variables that satisfies every inequality.
  - It is unsatisfiable otherwise.

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, J_4) = 7$

Phases:

- Find a suitable constraint satisfaction paradigm.
- Devise a formula that represents the combinatorial problem in the selected paradigm.
- Encode the problem as a formula.
- Use a solver for the selected paradigm.
  - If satisfiable: decode the satisfying assignment.
  - If unsatisfiable: provide a proof of unsatisfiability.

Encoding Combinatorial Problems as SAT

Running Example

The formula we need is:

$$
\begin{aligned}
F_N = & \forall\ i < j < k < N. \\
& -1x_{i,j} + -1x_{j,k} + -1x_{i,k} \geq -2 \\
& \forall\ i_1 < i_2 < i_3 < i_4 < N. \\
& 1x_{i_1,i_2} + 1x_{i_1,i_3} + 1x_{i_1,i_4} + 1x_{i_2,i_3} + 1x_{i_2,i_4} + 1x_{i_3,i_4} \geq 2
\end{aligned}
$$

F_N is satisfiable if and only if K_N can be colored in a way that avoids K_3 in the first color and J_4 in the second color.

Encoding Combinatorial Problems as SAT

Running Example

Theorem: $R(K_3, J_4) = 7$

Phases:

- Find a suitable constraint satisfaction paradigm.
- Devise a formula that represents the combinatorial problem in the selected paradigm.
- Encode the problem as a formula.
- Use a solver for the selected paradigm.
  - If satisfiable: decode the satisfying assignment.
  - If unsatisfiable: provide a proof of unsatisfiability.

Thanks!