Outline

Binary Resolution Inference System

The binary resolution inference system, denoted by $\mathbb{BR}$, consists of two inference rules:

- Binary resolution, denoted by BR

$$\frac{p \vee C_1 \quad \neg p \vee C_2}{C_1 \vee C_2} \text{ (BR).}$$

- Factoring, denoted by Fact:

$$\frac{L \vee L \vee C}{L \vee C} \text{ (Fact).}$$

Binary Resolution Inference System

The binary resolution inference system, denoted by $\mathbb{BR}$, consists of two inference rules:

- Binary resolution, denoted by BR

$$\frac{p \vee C_1 \quad \neg p \vee C_2}{C_1 \vee C_2} \text{ (BR).}$$

- Factoring, denoted by Fact:

$$\frac{L \vee L \vee C}{L \vee C} \text{ (Fact).}$$

Binary Resolution Inference System

The binary resolution inference system, denoted by $\mathbb{BR}$, consists of two inference rules:

- Binary resolution, denoted by BR

$$\frac{p \vee C_1 \quad \neg p \vee C_2}{C_1 \vee C_2} \text{ (BR)}.$$

- Factoring, denoted by Fact:

$$\frac{L \vee L \vee C}{L \vee C} \text{ (Fact)}.$$

Inference System

- inference has the form

$$\frac{F_1 \quad \ldots \quad F_n}{G},$$

where $n \geq 0$ and $F_1, \ldots, F_n, G$ are formulas.
- The formula G is called the conclusion of the inference;
- The formulas $F_1, \ldots, F_n$ are called its premises.
- An inference rule R is a set of inferences.
- Every inference $I \in R$ is called an instance of R.
- An Inference system $\mathbb{I}$ is a set of inference rules.
- Axiom: inference rule with no premises.

Inference System

- inference has the form

$$\frac{F_1 \quad \ldots \quad F_n}{G},$$

  where $n \geq 0$ and $F_1, \ldots, F_n, G$ are formulas.
- The formula G is called the conclusion of the inference;
- The formulas $F_1, \ldots, F_n$ are called its premises.
- An inference rule R is a set of inferences.
- Every inference $I \in R$ is called an instance of R.
- An Inference system $\mathbb{I}$ is a set of inference rules.
- Axiom: inference rule with no premises.

Inference System

- inference has the form

$$\frac{F_1 \quad \ldots \quad F_n}{G},$$

  where $n \geq 0$ and $F_1, \ldots, F_n, G$ are formulas.
- The formula G is called the conclusion of the inference;
- The formulas $F_1, \ldots, F_n$ are called its premises.
- An inference rule R is a set of inferences.
- Every inference $I \in R$ is called an instance of R.
- An Inference system $\mathbb{I}$ is a set of inference rules.
- Axiom: inference rule with no premises.

Inference System

- inference has the form

$$\frac{F_1 \quad \ldots \quad F_n}{G},$$

  where $n \geq 0$ and $F_1, \ldots, F_n, G$ are formulas.
- The formula G is called the conclusion of the inference;
- The formulas $F_1, \ldots, F_n$ are called its premises.
- An inference rule R is a set of inferences.
- Every inference $I \in R$ is called an instance of R.
- An Inference system $\mathbb{I}$ is a set of inference rules.
- Axiom: inference rule with no premises.

Derivation, Proof

- Derivation in an inference system $\mathbb{I}$: a tree built from inferences in $\mathbb{I}$.
- If the root of this derivation is E, then we say it is a derivation of E.
- Proof of E: a finite derivation whose leaves are axioms.
- Derivation of E from $E_1, \ldots, E_m$: a finite derivation of E whose every leaf is either an axiom or one of the expressions $E_1, \ldots, E_m$.

Soundness

- An inference is sound if the conclusion of this inference is a logical consequence of its premises.
- An inference rule is sound if every inference of this rule is sound.
- An inference system is sound if every inference rule in this system is sound.

Theorem

$\mathbb{BR}$ *is sound.*

Soundness

- An inference is sound if the conclusion of this inference is a logical consequence of its premises.
- An inference rule is sound if every inference of this rule is sound.
- An inference system is sound if every inference rule in this system is sound.

Theorem
$\mathbb{BR}$ *is sound.*

Consequence of Soundness

Theorem

Let S be a set of clauses. If $\Box$ can be derived from S in $\mathbb{BR}$, then S is unsatisfiable.

Example

Consider the following set of clauses

$$\{\neg p \vee \neg q,\ \neg p \vee q,\ p \vee \neg q,\ p \vee q\}.$$

The following derivation derives the empty clause from this set:

$$\dfrac{\dfrac{\dfrac{p \vee q \quad p \vee \neg q}{p \vee p}\,(\mathrm{BR})}{p}\,(\mathrm{Fact}) \qquad \dfrac{\dfrac{\dfrac{\neg p \vee q \quad \neg p \vee \neg q}{\neg p \vee \neg p}\,(\mathrm{BR})}{\neg p}\,(\mathrm{Fact})}{}}{\Box}\,(\mathrm{BR})$$

Hence, this set of clauses is unsatisfiable.

Example

Consider the following set of clauses

$$\{\neg p \vee \neg q,\ \neg p \vee q,\ p \vee \neg q,\ p \vee q\}.$$

The following derivation derives the empty clause from this set:

$$\dfrac{\dfrac{\dfrac{p \vee q \quad p \vee \neg q}{p \vee p}\,(\mathrm{BR})}{p}\,(\mathrm{Fact}) \qquad \dfrac{\dfrac{\neg p \vee q \quad \neg p \vee \neg q}{\neg p \vee \neg p}\,(\mathrm{BR})}{\neg p}\,(\mathrm{Fact})}{\square}\,(\mathrm{BR})$$

Hence, this set of clauses is unsatisfiable.

Example

Consider the following set of clauses

$$\{\neg p \vee \neg q,\ \neg p \vee q,\ p \vee \neg q,\ p \vee q\}.$$

The following derivation derives the empty clause from this set:

$$\dfrac{\dfrac{\dfrac{p \vee q \quad p \vee \neg q}{p \vee p}\ (\mathrm{BR})}{p}\ (\mathrm{Fact}) \qquad \dfrac{\dfrac{\neg p \vee q \quad \neg p \vee \neg q}{\neg p \vee \neg p}\ (\mathrm{BR})}{\neg p}\ (\mathrm{Fact})}{\Box}\ (\mathrm{BR})$$

Hence, this set of clauses is unsatisfiable.

Writing derivations in the linear form

(1)	$\neg p \vee \neg q$	input	
(2)	$\neg p \vee q$	input	
(3)	$p \vee \neg q$	input	
(4)	$p \vee q$	input	
(5)	$\neg p \vee \neg p$	BR	(1,2)
(6)	$\neg p$	Fact	(5)
(7)	$p \vee p$	BR	(3,4)
(8)	p	Fact	(7)
(9)	$\square$	BR	(6,8)

Completeness

$\mathbb{BR}$ is complete, that is, if a set of clauses is unsatisfiable, then one can derive an empty clause from this set.

Selection Function

The binary resolution inference system has too many inferences. There are restrictions on resolution that allow for fewer inferences but preserve completeness.

To define these systems we need a new notion.

A literal selection function selects one or more literals in every non-empty clause.

We will sometimes denote selected literals by underlining them, e.g.,

$$\underline{p} \lor \neg q$$

Selection Function

The binary resolution inference system has too many inferences. There are restrictions on resolution that allow for fewer inferences but preserve completeness.

To define these systems we need a new notion.

A literal selection function selects one or more literals in every non-empty clause.

We will sometimes denote selected literals by underlining them, e.g.,

$$\underline{p} \vee \neg q$$

Selection Function

The binary resolution inference system has too many inferences. There are restrictions on resolution that allow for fewer inferences but preserve completeness.

To define these systems we need a new notion.

A literal selection function selects one or more literals in every non-empty clause.

We will sometimes denote selected literals by underlining them, e.g.,

$$\underline{p} \vee \neg q$$

Selection Function

The binary resolution inference system has too many inferences. There are restrictions on resolution that allow for fewer inferences but preserve completeness.

To define these systems we need a new notion.

A literal selection function selects one or more literals in every non-empty clause.

We will sometimes denote selected literals by underlining them, e.g.,

$$\underline{p} \lor \neg q$$

Binary Resolution with Selection

The binary resolution inference system, denoted by $\mathbb{BR}_\sigma$, consists of two inference rules:

- Binary resolution denoted by BR

$$\frac{\underline{p} \vee C_1 \quad \underline{\neg p} \vee C_2}{C_1 \vee C_2} \text{ (BR)}.$$

- Factoring, denoted by Fact:

$$\frac{\underline{L} \vee \underline{L} \vee C}{L \vee C} \text{ (Fact)}.$$

Binary resolution with selection is incomplete.

However, it is complete for some well-behaved selection functions.

Binary Resolution with Selection

The binary resolution inference system, denoted by $\mathbb{BR}_\sigma$, consists of two inference rules:

- Binary resolution denoted by BR

$$\frac{\underline{p} \vee C_1 \quad \underline{\neg p} \vee C_2}{C_1 \vee C_2} \text{ (BR).}$$

- Factoring, denoted by Fact:

$$\frac{\underline{L} \vee L \vee C}{L \vee C} \text{ (Fact).}$$

Binary resolution with selection is incomplete.

However, it is complete for some well-behaved selection functions.

Binary Resolution with Selection

The binary resolution inference system, denoted by $\mathbb{BR}_\sigma$, consists of two inference rules:

- Binary resolution denoted by BR

$$\frac{\underline{p} \vee C_1 \quad \underline{\neg p} \vee C_2}{C_1 \vee C_2} \text{ (BR)}.$$

- Factoring, denoted by Fact:

$$\frac{\underline{L} \vee L \vee C}{L \vee C} \text{ (Fact)}.$$

Binary resolution with selection is incomplete.

However, it is complete for some well-behaved selection functions.

Unrestricted binary resolution and binary resolution with selection

Consider the selection function that selects all literals in a clause. Then the binary resolution rule:

$$\frac{p \vee C_1 \quad \neg p \vee C_2}{C_1 \vee C_2} \text{ (BR).}$$

becomes a special case of binary resolution with selection.

Literal Orderings

Consider any total ordering $\succ$ on propositional variables. We want to extend it to literals.

Let $L_1 = (\neg)A_1$ and $L_2 = (\neg)A_2$ be literals. We let $L_1 \succ_{lit} L_2$ if and only if one of the following conditions holds:

1. $A_1 \succ A_2$; or
2. $A_1 = A_2$, L_1 is negative and L_2 is positive.

In other words, we compare literals by first comparing the atoms of these literals and if the atoms are equal define the negative literal to be greater.

Literal Orderings

Consider any total ordering $\succ$ on propositional variables. We want to extend it to literals.

Let $L_1 = (\neg)A_1$ and $L_2 = (\neg)A_2$ be literals. We let $L_1 \succ_{lit} L_2$ if and only if one of the following conditions holds:

1. $A_1 \succ A_2$; or
2. $A_1 = A_2$, L_1 is negative and L_2 is positive.

In other words, we compare literals by first comparing the atoms of these literals and if the atoms are equal define the negative literal to be greater.

Literal Orderings

Consider any total ordering $\succ$ on propositional variables. We want to extend it to literals.

Let $L_1 = (\neg)A_1$ and $L_2 = (\neg)A_2$ be literals. We let $L_1 \succ_{lit} L_2$ if and only if one of the following conditions holds:

1. $A_1 \succ A_2$; or
2. $A_1 = A_2$, L_1 is negative and L_2 is positive.

In other words, we compare literals by first comparing the atoms of these literals and if the atoms are equal define the negative literal to be greater.

Ordered resolution

Fix an ordering $\succ$ on the set of propositional variables and let $\succ_{lit}$ be corresponding literal ordering. Consider the selection function σ that selects all maximal w.r.t. $\succ_{lit}$ literals.

Theorem

$\mathbb{BR}_\sigma$ *is complete, that is, for every unsatisfiable set of clauses S one can derive the empty clause from clauses in S using inferences in* $\mathbb{BR}_\sigma$.

Ordered resolution

Fix an ordering $\succ$ on the set of propositional variables and let $\succ_{lit}$ be corresponding literal ordering. Consider the selection function σ that selects all maximal w.r.t. $\succ_{lit}$ literals.

Theorem

$\mathbb{BR}_\sigma$ is complete, that is, for every unsatisfiable set of clauses S one can derive the empty clause from clauses in S using inferences in $\mathbb{BR}_\sigma$.

Ordered resolution: example

Assume $q \succ p$.

(1)	$\neg p \vee \underline{\neg q}$	input	
(2)	$\neg p \vee \underline{q}$	input	
(3)	$p \vee \underline{\neg q}$	input	
(4)	$p \vee \underline{q}$	input	
(5)	$\underline{\neg p} \vee \underline{\neg p}$	BR	(1,2)
(6)	$\underline{p} \vee \underline{p}$	BR	(3,4)
(7)	$\underline{p}$	Fact	(6)
(8)	$\underline{\neg p}$	BR	(6,7)
(9)	$\square$	BR	(6,8)

Note: fewer inferences are enabled compared to unrestricted binary resolution.

Ordered resolution: example

Assume $q \succ p$.

(1)	$\neg p \vee \underline{\neg q}$	input	
(2)	$\neg p \vee \underline{q}$	input	
(3)	$p \vee \underline{\neg q}$	input	
(4)	$p \vee \underline{q}$	input	
(5)	$\underline{\neg p} \vee \underline{\neg p}$	BR	(1,2)
(6)	$\underline{p} \vee \underline{p}$	BR	(3,4)
(7)	$\underline{p}$	Fact	(6)
(8)	$\underline{\neg p}$	BR	(6,7)
(9)	$\square$	BR	(6,8)

Note: fewer inferences are enabled compared to unrestricted binary resolution.

Inference Process

Inference process: sequence of sets of clauses $S_0, S_1, \ldots$, denoted by

$$S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$$

$(S_i \Rightarrow S_{i+1})$ is a step of this process.

We say that this step is an $\mathbb{I}$-step if

1. there exists an inference

$$\frac{C_1 \quad \ldots \quad C_n}{C}$$

   in $\mathbb{I}$ such that $\{C_1, \ldots, C_n\} \subseteq S_i$;
2. $S_{i+1} = S_i \cup \{C\}$.

An $\mathbb{I}$-inference process is an inference process whose every step is an $\mathbb{I}$-step.

Inference Process

Inference process: sequence of sets of clauses $S_0, S_1, \ldots$, denoted by

$$S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$$

$(S_i \Rightarrow S_{i+1})$ is a step of this process.

We say that this step is an $\mathbb{I}$-step if

1. there exists an inference

$$\frac{C_1 \quad \ldots \quad C_n}{C}$$

   in $\mathbb{I}$ such that $\{C_1, \ldots, C_n\} \subseteq S_i$;
2. $S_{i+1} = S_i \cup \{C\}$.

An $\mathbb{I}$-inference process is an inference process whose every step is an $\mathbb{I}$-step.

Inference Process

Inference process: sequence of sets of clauses $S_0, S_1, \ldots$, denoted by

$$S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$$

$(S_i \Rightarrow S_{i+1})$ is a step of this process.

We say that this step is an $\mathbb{I}$-step if

1. there exists an inference

$$\frac{C_1 \quad \ldots \quad C_n}{C}$$

   in $\mathbb{I}$ such that $\{C_1, \ldots, C_n\} \subseteq S_i$;
2. $S_{i+1} = S_i \cup \{C\}$.

An $\mathbb{I}$-inference process is an inference process whose every step is an $\mathbb{I}$-step.

Property

Lemma

Let $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$ be an $\mathbb{I}$-inference process and a clause C belongs to some S_i. Then S_i is derivable in $\mathbb{I}$ from S_0.

Can we prove the inverse?

Property

Lemma

Let $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$ *be an* $\mathbb{I}$*-inference process and a clause* C *belongs to some* S_i*. Then* S_i *is derivable in* $\mathbb{I}$ *from* S_0*.*

Can we prove the inverse?

Limit and Fairness

The limit of an inference process $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$ is the set of clauses $\bigcup_i S_i$.

Let $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$ be an inference process with the limit S_∞. The process is called fair if for every $\mathbb{I}$-inference

$$\frac{C_1 \quad \ldots \quad C_n}{C},$$

if $\{C_1, \ldots, C_n\} \subseteq S_\infty$, then there exists i such that $C \in S_i$.

Limit and Fairness

The limit of an inference process $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \dots$ is the set of clauses $\bigcup_i S_i$.

Let $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \dots$ be an inference process with the limit S_∞. The process is called fair if for every $\mathbb{I}$-inference

$$\frac{C_1 \quad \dots \quad C_n}{C} \; ,$$

if $\{C_1, \dots, C_n\} \subseteq S_\infty$, then there exists i such that $C \in S_i$.

Completeness, reformulated

Theorem Let $\mathbb{I}$ be an inference system. The following conditions are equivalent.

1. $\mathbb{I}$ is complete.
2. For every unsatisfiable set of clauses S_0 and any fair $\mathbb{I}$-inference process with the initial set S_0, the limit of this inference process contains $\square$.

Saturated Set of Clauses

Let $\mathbb{I}$ be an inference system and S be a set of clauses. S is called saturated with respect to $\mathbb{I}$, or simply $\mathbb{I}$-*saturated*, if for every inference of $\mathbb{I}$ with premises in S, the conclusion of this inference also belongs to S.

The closure of S with respect to $\mathbb{I}$, or simply $\mathbb{I}$-closure, is the smallest set S' containing S and saturated with respect to $\mathbb{I}$.

Completeness of Ordered Resolution

Theorem (Completeness)

Take any well-founded ordering $\succ$ and consider the selection function σ that selects all maximal w.r.t. $\succ_{lit}$ literals. Let S_0 be a set of clauses and $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$ be a fair $\mathbb{BR}_\sigma$-inference process. Then S_0 is unsatisfiable if and only if $\square \in S_i$ for some i.

Lemma

The limit S_ω is saturated.

Lemma

The limit S_ω is logically equivalent to the initial set S_0.

Lemma

A saturated set S of clauses is unsatisfiable if and only if $\square \in S$.

Completeness of Ordered Resolution

Theorem (Completeness)

Take any well-founded ordering $\succ$ and consider the selection function σ that selects all maximal w.r.t. $\succ_{lit}$ literals. Let S_0 be a set of clauses and $S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$ be a fair $\mathbb{BR}_\sigma$-inference process. Then S_0 is unsatisfiable if and only if $\square \in S_i$ for some i.

Lemma

The limit S_ω is saturated.

Lemma

The limit S_ω is logically equivalent to the initial set S_0.

Lemma

A saturated set S of clauses is unsatisfiable if and only if $\square \in S$.

Corollaries

Completeness of Binary Resolution. Binary resolution is complete.
Compactness. Let S be a countably infinite set of clauses. Then S is unsatisfiable if and only if it contains a finite unsatisfiable subset.
Note. The assumption of being countably infinite can be dropped.

Problem: search space grows too fast

Idea: remove some clauses from the search space.
We will consider later how clauses can be removed without compromising completeness.

Inference Process with Deletion

Let $\mathbb{I}$ be an inference system. Consider an inference process with two kinds of step $S_i \Rightarrow S_{i+1}$:

1. $\mathbb{I}$-inference;
2. deletion of a clause in S_i, that is

$$S_{i+1} = S_i - \{C\},$$

   where $C \in S_i$.

Fairness: Persistent Clauses and Limit

Consider an inference process

$$S_0 \Rightarrow S_1 \Rightarrow S_2 \Rightarrow \ldots$$

A clause C is called persistent if

$$\exists i \forall j \geq i (C \in S_j).$$

The limit S_ω of the inference process is the set of all persistent clauses:

$$S_\omega = \bigcup_{i=0,1,\ldots} \bigcap_{j \geq i} S_j.$$

Fairness

The process is called $\mathbb{I}$-fair if every inference with persistent premises in S_ω has been applied, that is, if

$$\frac{C_1 \quad \ldots \quad C_n}{C}$$

is an inference in $\mathbb{I}$ and $\{C_1, \ldots, C_n\} \subseteq S_\omega$, then $C \in S_i$ for some i.

Deletion rules

Tautology: a clause of the form $p \vee \neg p \vee C$. Tautology deletion: deletion of tautologies from the search space.

Finite multiset: like a set but elements may occur more than once. Example: $\{1, 2, 2, 5, 5, 5\}$. A clause can be considered as a multiset of its literals.

A clause C_1 is said to subsume any clause $C_1 \vee C_2$, where C_2 is non-empty. In other words, C_1 subsumes C_2 if and only if C_1 is a submultiset of C_2.

Subsumption deletion: deletion of subsumed clauses from the search space.

Completeness with deletion rules

Subsumption and tautology deletion does not compromise completeness of binary and ordered resolution.
That is, for every fair inference process with subsumption a tautology deletion, if the initial set of clauses is unsatisfiable, then the limit of the process contains the empty clause.

Example: inference process with deletion

(1)	$\neg p \vee \underline{\neg q}$	input	
(2)	$\neg p \vee \underline{q}$	input	
(3)	$p \vee \underline{\neg q}$	input	
(4)	$p \vee \underline{q}$	input	

(1)	$\neg p \vee \underline{\neg q}$	input	
(2)	$\neg p \vee \underline{q}$	input	
(3)	$p \vee \underline{\neg q}$	input	
(4)	$p \vee \underline{q}$	input	
(5)	$\underline{\neg p} \vee \underline{\neg p}$	BR	(1,2)

(1)	$\neg p \vee \underline{\neg q}$	input	
(2)	$\neg p \vee \underline{q}$	input	
(3)	$p \vee \underline{\neg q}$	input	
(4)	$p \vee \underline{q}$	input	
(5)	$\underline{\neg p} \vee \underline{\neg p}$	BR	(1,2)
(6)	$\underline{\neg p}$	Fact	(5)

(3)	$p \vee \underline{\neg q}$	input	
(4)	$p \vee \underline{q}$	input	
(6)	$\underline{\neg p}$	Fact	(5)

Example: inference process with deletion

(3)	$p \lor \underline{\neg q}$	input	
(4)	$p \lor \underline{q}$	input	
(6)	$\underline{\neg p}$	Fact	(5)

(3)	$p \lor \underline{\neg q}$	input	
(4)	$p \lor \underline{q}$	input	
(6)	$\underline{\neg p}$	Fact	(5)
(7)	$\underline{p} \lor \underline{p}$	BR	(3,4)

(3)	$p \lor \underline{\neg q}$	input	
(4)	$p \lor \underline{q}$	input	
(6)	$\underline{\neg p}$	Fact	(5)
(7)	$\underline{p} \lor \underline{p}$	BR	(3,4)
(8)	$\underline{p}$	Fact	(7)

(6)	$\underline{\neg p}$	Fact	(5)
(8)	$\underline{p}$	Fact	(7)

(6)	$\underline{\neg p}$	Fact	(5)
(8)	$\underline{p}$	Fact	(7)
(9)	$\square$	BR	(6,8)