Estratégias de resolução

1 Estratégias

Um problema com uma aplicação não restringida do algoritmo de resolução é que a cada passo têm várias possibilidades de aplicação da regra de resolução. Podemos tentar todas as possiblidades. Assim temos certeza de obter a cláusula vazia, se for possível produzi-la. Mas teremos que enfrentar o problema da explosão combinatória que tornará inviável qualquer implementação. Então, o uso eficiente do algoritmo de resolução é semelhante a um problema de busca. Existem algumas estratégias para tornar mais eficiente o algoritmo de resolução que são discutidas nessa seção. Para simplificar, as estratégias são apresentadas com exemplos da resolução proposicional, mas ela se aplicam também à líogica de primeira ordem.

1.1 Produção da cláusula vazia

Para entender as principais estratégias de resolução, e preciso primeiro entender um teorema importante:

Teorema 1 Se existe uma derivação da cláusula vazio a partir de uma cláusula $C = [X_1, X_2, \ldots, X_n]$, existe uma derivação que elimina sequencialmente cada literal de C para produzir a cláusula vazia.

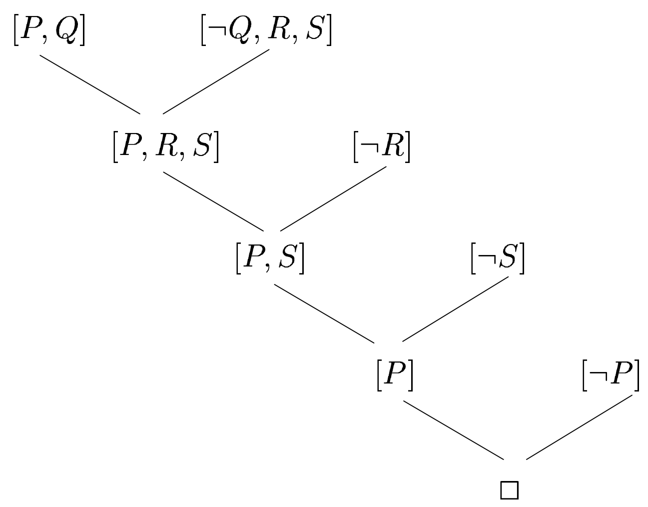
Prova: Seja $C = [X_1, X_2, \ldots, X_n]$. Por hipótese, $C \vdash \Box$. Para obter a cláusula vazia, necessariamente em uma etapa da derivação, uma cláusula descendente de C é resolvida com uma cláusula $[\neg X_n, Y_1, \ldots, Y_k]$. O resultado será uma cláusula $C_i = [\ldots, Y_1, \ldots, Y_k]$. Como a derivação chega à cláusula vazia, existe também uma maneira de eliminar $Y_1, \ldots, Y_k$. A eliminação de um literal de uma cláusula não depende dos outros literais que ela contém. Se é possível eliminar $Y_1, \ldots, Y_k$ da cláusula C_i, é possível eliminar $Y_1, \ldots, Y_k$ da cláusula $[\neg X_n, Y_1, \ldots, Y_k]$ e obter $[\neg X_n]$. Portanto é possivel eliminar X_n de C e obter a cláusula $[X_1, X_2, \ldots, X_{n-1}]$. Isso é verdadeiro para qualquer literal de C, isto é, para cada literal de C existe a possibilidade de produzir uma cláusula que contém somente o complemento desse literal. Portanto, é possível eliminar sequencialmente todos os literais de C, e em qualquer ordem. ■

Sabemos que para obter a cláusula vazia, devemos identificar uma derivação que contém no mínimo uma cláusula do conjunto inicial. De acordo com o teorema 1, podemos afirmar que a obtenção da cláusula vazia consiste em eliminar sequencialmente todos os literais de uma cláusula do conjunto inicial.

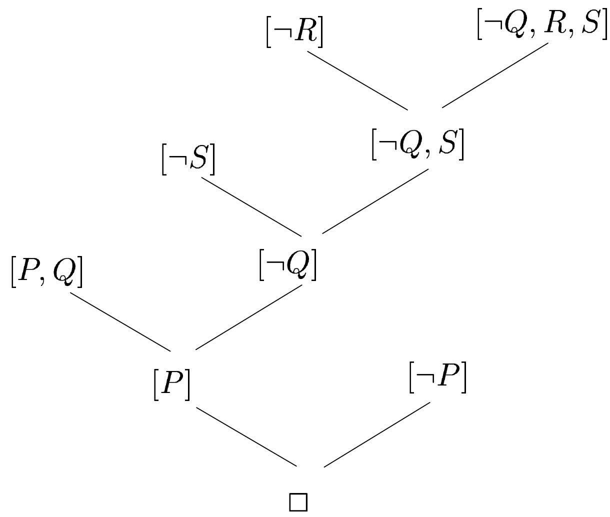
Sabemos que as vezes a resolução produz uma cláusula resultante que contém mais literais que cada uma da duas cláusula resolvidas. Suponhamos por exemplo o seguinte conjunto de fórmulas:

(1) $[P, Q]$
(2) $[\neg Q, R, S]$
(3) $[\neg R]$
(4) $[\neg S]$
(5) $[\neg P]$

A partir da cláusula $[P, Q]$, podemos produzir a cláusula vazia com a seguinte derivação:



Para obter a claúsula vazia, o tamanho aumentou para diminuir depois. De acordo com o teorema 1 podemos, com o mesmo conjunto inicial de cláusulas, produzir a cláusula vazia, onde os literais da cláusula $[P, Q]$ são eliminados sequencialmente:



1.2 Algoritmo genérico

Antes de discutir as estratégias, vamos ver o comportamento do algoritmo no caso onde todas as possibilidades são tentadas. Eis o algoritmo:

Resolução genérica: Suponhamos dois conjuntos: S, o conjunto atualizado de fórmulas, e S' o conjunto das novas fórmulas obtidas no último passo. Inicialmente, $S = \Delta$ e $S' = \emptyset$. Chamamos duas cláusulas *candidatas* se uma contém um literal l e a outra o literal $\neg l$.

Enquanto $\square \notin S'$:
$S \leftarrow S \cup S'$
$S' = \{r | s_1 \in S', s_2 \in S \cup S', r = \text{resolvente de } s_1 \text{ e } s_2\}$

Eis a prova obtida usando esse algoritmo com o exemplo da seção **??**:

(1)	$[P, Q]$
(2)	$[P, R]$
(3)	$[\neg Q, \neg R]$
(4)	$[\neg P]$

(5)	$[P, \neg R]$	(1,3)
(6)	$[Q]$	(1,4)
(7)	$[P, \neg Q]$	(2,3)
(8)	$[R]$	(2,4)
(9)	$[P]$	(1,7)
(10)	$[P]$	(2,5)
(11)	$[\neg R]$	(3,6)
(12)	$[\neg Q]$	(3,8)
(13)	$[\neg R]$	(4,5)
(14)	$[\neg Q]$	(4,7)
(15)	$[P]$	(1,12)
(16)	$[P]$	(1,14)
(17)	$[P]$	(2,11)
(18)	$[P]$	(2,13)
(19)	$\square$	(4,9)
(20)	$\square$	(4,10)
(21)	$\square$	(6,12)
(22)	$\square$	(6,14)
(23)	$\square$	(8,11)
(24)	$\square$	(8,13)

Podemos ver que as mesmas fórmulas são reproduzidas várias vezes inutilmente. Mas o problema principal é que obtemos uma prova bem mais comprida. O algoritmo pode ser melhorado se não esperamos que todas as novas fórmulas sejam produzidas antes de testar a existência da cláusula vazia. Outra melhoria seria a eliminação das redundâncias depois de cada passo. Entretanto, essas modificações não vão eliminar o problema da explosão combinatória.

1.3 Estratégias de eliminação

1.3.1 Simplificação

Na aplicação do algoritmo de resolução proposicional, pode aparecer uma cláusula que contém um mesmo literal repetido. Nesse caso, podemos conservar somente um dele e obter uma cláusula equivalente. Por exemplo, a clausula $[P, \neg Q, W, P, \neg Q]$ é equivalente a $[P, \neg Q, W]$.

Essa simplificação é as vezes necessária para poder obter a cláusula vazia. Isso significa que a resolução não é completa se não efetuamos essa simplificação. Vamos mostrar isso com um exemplo. Seja as seguintes cláusulas:

(1) $[P, P]$
(2) $[\neg P, \neg P]$

Não é difícil conferir que esse conjunto é inconsistente. Aplicando a resolução com essas fórmulas, não obteremos a cláusula vazia:

(1)	$[P, P]$	
(2)	$[\neg P, \neg P]$	
(3)	$[P, \neg P]$	(1,2)
(4)	$[P, P]$	(1,3)
(5)	$[\neg P, \neg P]$	(2,3)
	...	

Após simplificação, obtemos a cláusula vazia imediatamente:

(1) $[P]$
(2) $[\neg P]$
(3) $\square$

1.3.2 Fatorização

No caso da lógica de primeira ordem, a situação é mais complicada. Não é correto eliminar a repetição de um predicado. Por exemplo, a seguinte cláusula $[\mathrm{p}(x), \mathrm{p}(\mathrm{A}), \mathrm{q}(y)]$ não pode ser simplificada para obter a cláusula $[\mathrm{p}(x), \mathrm{q}(y)]$. Se rescrevemos essas duas cláusulas na sintaxe da lógica de primeira ordem, obtemos essas duas fórmulas, que são claramente não equivalentes:

$$\forall x \forall y [\mathrm{p}(x) \vee \mathrm{p}(\mathrm{A}) \vee \mathrm{q}(y)]$$
$$\forall y [\mathrm{p}(\mathrm{A}) \vee \mathrm{q}(y)]$$

Considere agora essas duas cláusulas:

$$[\mathrm{p}(x), \mathrm{p}(\mathrm{f}(y))]$$
$$[\neg \mathrm{p}(w), \neg \mathrm{p}(z)]$$

Essas cláusulas são inconsistente. para ver isso, é só substituir x, w e z por $\mathrm{f}(y)$, respectivamente. Nesse caso, a primeira cláusula é equivalente a $\mathrm{p}(\mathrm{f}(y))$ e a segunda, $\neg \mathrm{p}(\mathrm{f}(y))$. Infelizmente, o algoritmo de resolução como definido não permite produzir a cláusula vazia a partir dessas cláusulas. Para conseguir isso, temos que modificar a regra de resolução:

Seja C_1, C_2 duas cláusulas. Seja também C_1', C_2' subconjuntos de C_1 e C_2 respectivamente tais que todos os literais de C_1' podem ser unificados com as negações de todos os literais de C_2', usando o unificador μ. Logo essas duas cláusulas tem um resolvente ρ obtido da seguinte maneira:

$$\rho = \mathrm{SUBST}(\mu, [(C_1 - C_1') \cup (C_2 - C_2')])$$

1.3.3 Remoção das tautologias

Uma tautologia não adianta nada na resolução. Para entender isso, lembramos que uma tautologia, na forma disjuntiva normal, é uma cláusula que contém um literal e a negação desse literal. Supohnamos tal cláusula, seja $[P, \neg P, Q]$. Supohnamos que é possível obter a cláusula vazia a partir dessa cláusula. Isso significa que é possível eliminar sucessivamente cada literal da cláusula. Consideramos agora a situação onde Q foi eliminado, obtendo $[P, \neg P]$. Agora, para obter a cláusula vazia, deveremos resolver primeiro com $[P]$ e depois com $[\neg P]$ (ou o contrário). Isso significa que essas duas cláusulas devem existir no conjunto. Se elas já existem, não é preciso utilizar $[Q, P, \neg P]$ para produzir a cláusula vazia, pois essas duas cláusulas sozinhas permitem isso.

Existe uma outra maneira de entender porque podemos eliminar as tautologias do conjunto de cláusulas. Queremos provar que o conjunto é inconsistente. Já sabemos que $P \wedge \top \equiv P$. Isso significa que o valor de verdade de P é o mesmo com ou sem a tautologia. Portanto, se ele é inconsistente com a tautologia, ele continua inconsistente sem ela.

Na lógica de primeira ordem, é preciso ser cauteloso na eliminação de tautologias. Eis exemplos de cláusulas que são tautologias:

$[\mathrm{p}(x), \neg\mathrm{p}(x)]$

Mais cuidado com essas cláusulas que não são tautologias:

$[\mathrm{p}(x), \neg\mathrm{p}(\mathrm{A})]$
$[\mathrm{p}(x), \neg\mathrm{p}(y)]$

A primeira fórmula é equivalente a $\forall x[\mathrm{p}(\mathrm{A}) \supset \mathrm{p}(x)]$ que evidentemente não é uma tautologia. A segunda é equivalente a $\forall x \mathrm{p}(x) \vee \forall y \neg\mathrm{p}(y)$. Isso significa que ou todo mundo tem a propriedade p ou todo mundo não tem essa propriedade. Isso não é uma tautologia, pois podemos imaginar um mundo onde alguns têm a propriedade e alguns não têm.

Como uma tautologia poder aparecer em qualquer momento na execução do algoritmo, devemos tester, cada vez que duas cláusulas são resolvidas, se a resolvente é uma tautologia.

1.3.4 Remoção de cláusula com literal puro

Um literal é dito **puro** se no conjunto de cláusulas onde ele se encontra não existe o seu complemento. Se no conjunto de cláusulas aperecem literais puros, podemos eliminar toda cláusula que contém um literal puro. A razão disso é que essas cláusulas não poderão ser usadas para produzir a cláusula vazia, pois não tem jeito de eliminiar o literal puro. Todos os descendentes dessa cláusula vão conter o literal puro. Eis um exemplo:

(1)	$[\neg P, \neg Q, R]$	
(2)	$[\neg P, S]$	
(3)	$[\neg Q, S]$	
(4)	$[P]$	
(5)	$[Q]$	
(6)	$[\neg R]$	
(7)	$[\neg Q, R]$	(1,4)
(8)	$[\neg P, R]$	(1,5)
(9)	$[\neg P, \neg Q]$	(1,6)
(10)	$[S]$	(2,4)
(11)	$[S]$	(3,4)
(12)	$[R]$	(4,8)
(13)	$[\neg Q]$	(4,9)
(14)	$[R]$	(5,7)
(15)	$[\neg P]$	(5,9)
(16)	$[\neg Q]$	(6,7)
(17)	$[\neg P]$	(6,8)
(18)	$\square$	(4,15)
(19)	$\square$	(4,17)
(20)	$\square$	(5,13)
(21)	$\square$	(5,16)
(22)	$\square$	(6,12)
(23)	$\square$	(6,14)

Considerando todas as derivações que chegaram à cláusula vazia, observamos que em nenhuma dela aparecem as cláusulas (2) e (3), que contêm um literal puro. Portanto, podemos eliminá-las do conjunto.

Se um conjunto de fórmula não contém literais puros, não pode aparecer um literal puro no decorrer do algoritmo. Então, a deteção de literais puros é feita uma vez só, antes de começar a execução do algoritmo.

1.3.5 Remoção de subjugação

Seja C_1 e C_2 tal que todos os literais de C_1 se encontram em C_2. Dizemos que C_1 subjuga C_2. Uma coisa interessante é que quando C_1 é verdadeira C_2 também é. Uma consequência disso é que não precisamos de C_2 para efetuar a nossa prova.

Seja as cláusulas $C_1 = [P, Q]$ e $C_2 = [P, Q, R]$. A cláusula C_1 subjuga a cláusula C_1. Supohnamos que existe uma derivação para produzir a cláusula vazia a partir de C_2. Isso significa que existe uma derivação que retira da cláusula o literal R. Isso resulta em uma cláusula que já temos. Portanto, se existe uma derivação que usa C_2, existe uma que usa C_1, e não precisamos de de C_2.

Na lógica de primeira ordem, uma cláusula C subjuga uma outra cláusula D se existe uma substituição σ tal que $\mathrm{SUBST}(\sigma, C) \subseteq D$. Considere por exemplos as duas seguintes cláusulas:

1) $[\mathrm{q}(x), \neg\mathrm{p}(\mathrm{A})]$
2) $[\neg\mathrm{p}(y)]$

A segunda cláusula subjuga a primeira, pois com a substituição $\{y/\mathrm{A}\}$, obtemos uma cláusula que é um subconjunto da primeira.

Um problema com a subjugação é que ela tem que ser verificada com todas os pares de fórmulas e a cada passo na execução do algoritmo. Isso pode tornar o algoritmo muito ineficiente. Portanto, essa técnica não é muito usada.

2 Resolução unitária

Considerando que a cláusula vazia pode ser obtida por eliminação sequencial dos literais que ela contém, uma estratégia para obter rapidamente a cláusula vazia consiste em resolver sempre com uma cláusula unitária, isto é, que contém somente um literal. Assim, a cada passo do algoritmo, uma cláusula diminui de tamanho. Eventualmente deveriamos chegar à cláusula vazia.

A resolução unitária é muito eficiente. Com o exemplo da seção **??**, obtemos uma derivação bem mais curto que a obtida com o algoritmo genérico:

(1)	$[P, Q]$	
(2)	$[P, R]$	
(3)	$[\neg Q, \neg R]$	
(4)	$[\neg P]$	
(5)	$[Q]$	(1,4)
(6)	$[R]$	(2,4)
(7)	$[\neg R]$	(3,5)
(8)	$[\neg Q]$	(3,6)
(9)	$[P]$	(2,7)
(10)	$\square$	(6,7)
(11)	$[P]$	(1,8)
(12)	$\square$	(5,8)

O problema com a resolução unitária é que ela torna incompleto o algoritmo de resolução. Eis um conjunto de fórmulas inconsistentes com o qual não podemos produzir a cláusula vazia utilizando essa estratégia:

(1)	$[\neg P, Q]$
(2)	$[P, \neg Q]$
(3)	$[\neg P, \neg Q]$
(4)	$[P, Q]$

Eis uma prova que produz a cláusula vazia:

(1)	$[\neg P, Q]$	
(2)	$[P, \neg Q]$	
(3)	$[\neg P, \neg Q]$	
(4)	$[P, Q]$	
(5)	$[\neg P]$	(1,3)
(6)	$[P]$	(2,4)
(7)	$\square$	(5,6)

No caso particular das cláusulas de Horn, essa estratégia é completa.

Se ao invés de exigir que uma das cláusulas seja unitária relaxamos as restrições para tentar selecionar preferencialmente uma cláusula unitária, o algoritmo se torna menos eficiente, mas fica completo. Essa estratégia sozinha não é suficiente para viabilizar o algoritmo de resolução, mas combinada com outras estratégias, ela melhora muito o desempenho.

3 Resolução input e resolução linear

A estratégia de resolução input consiste simplesmente em sempre utilizar no mínimo uma cláusula que faz parte do conjunto inicial. Eis uma aplicação dessa estratégia ao nosso exemplo:

(1)	$[P, Q]$	
(2)	$[P, R]$	
(3)	$[\neg Q, \neg R]$	
(4)	$[\neg P]$	
(5)	$[P, \neg R]$	(1,3)
(6)	$[Q]$	(1,4)
(7)	$[P, \neg Q]$	(2,3)
(8)	$[R]$	(2,4)
(9)	$[P]$	(1,7)
(10)	$[P]$	(2,5)
(11)	$[\neg R]$	(3,6)
(12)	$[\neg Q]$	(3,8)
(13)	$[\neg R]$	(4,5)
(14)	$[\neg Q]$	(4,7)
(15)	$[P]$	(1,12)
(16)	$[P]$	(1,14)
(17)	$[P]$	(2,11)
(18)	$[P]$	(2,13)
(19)	$\square$	(4,9)
(20)	$\square$	(4,10)

No nosso exemplo, essa estratégia não parece muito eficiente. Além disso, acontece que ela não é completa. Considere de novo o exemplo que foi usado para mostrar a incompletude da resolução unitária:

(1)	$[\neg P, Q]$
(2)	$[P, \neg Q]$
(3)	$[\neg P, \neg Q]$
(4)	$[P, Q]$

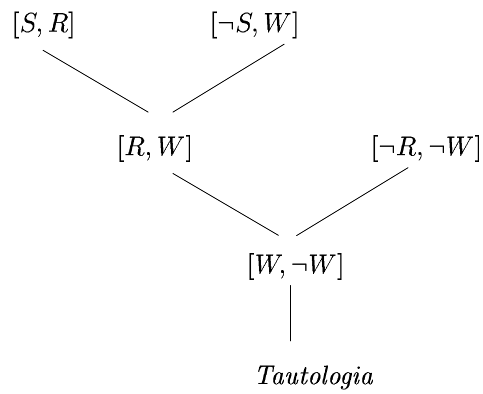
Nesse caso é claro que nunca conseguiremos produzir a cláusula vazia. É muito fácil entender porque, considerando que a obtenção da cláusula vazia exige duas cláusulas unitárias. Como não há nenhuma no conjunto inicial, não será possível obter a cláusula vazia.

De novo, no caso particular das cláusulas de Horn, essa estratégia é completa.

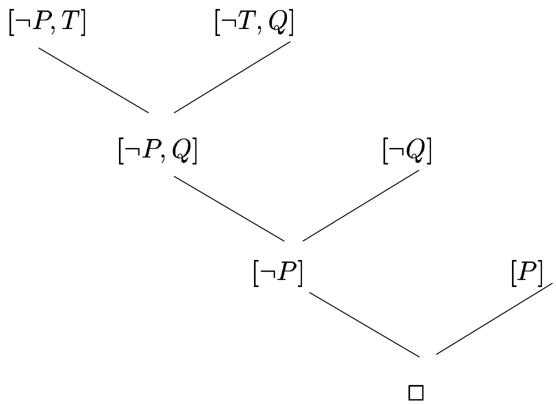
Existe uma generalização dessa estratégia, chamada **resolução linear**: duas cláusulas P e Q podem ser resolvidas somente se P faz parte do conjunto inicial o se ela é ancestral de Q na árvore de prova. Considere esse conjunto de fórmula:

(1)	$[S, R]$
(2)	$[\neg S, W]$
(3)	$[\neg R, \neg W]$
(4)	$[\neg P, T]$
(5)	$[\neg T, Q]$
(6)	$[P]$
(7)	$[\neg Q]$

A estratégia de resolução linear vai produzir a cláusula vazia, mas antes ela pode se perder em várias derivações inúteis. Por exemplo, se começamos pelas cláusulas (1) e (2), chegamos a uma situação sem saida:



Nesse caso é necessário recuar e ver se tem outras possibilidades. Tem uma outra possibilidade, onde as cláusulas (1) e (3) são resolvidas. Mas isso também leva a uma situação sem saida. Idem se recomeçamos com as cláusulas (2) e (3). Eventualemente, chegaremos à cláusula (4) que sim permite deduzir a cláusula vazia:



4 Conjunto de suporte

Seja o conceito de conjunto de suporte definido assim: o conjunto onde cada elemento é uma cláusula que resulta da transformação em forma normal do fato a provar que foi negado, ou um descendente de tal cláusula.

A estratégia é a seguinte: resolvemos duas cláusulas somente se no mínimo uma dela faz parte do conjunto de suporte. Para ilustrar essa estratégia, utilizaremos o mesmo exemplo que foi usada para explicar a resolução unitária. Supohnamos que o fato a provar era $\neg Q$. Então, o conjunto de suporte sera Q e todos os seus descendente. Eis a prova que obtemos: