Kapitola 4

Splniteľné modulové teórie

Absencia štruktúry

Na získanie vodičského oprávnenia skupiny B sa okrem úspešného vykonania skúšky požaduje minimálny vek v deň skúšky aspoň 17 rokov, pričom ak to bolo vo veku menej ako 18 rokov, tak je požadované pri vydávaní vodičského preukazu aj zaevidovanie osoby skúseného spolujazdca. Pri použití výrokovej logiky by táto podmienka mohla byť reprezentovaná napríklad v tvare zloženej vety

$$skuska \land \neg vek\_pod\_17 \land (vek\_pod\_18 \to spolujazdec)$$

kde štyri použité symboly sú najmenšími stavebnými jednotkami, ktoré sú interpretované ako pravdivé alebo nepravdivé. Pri hľadaní vhodnej interpretácie symbolov sa vychádza iba z faktu, či daná interpretácia symbolov reprezentuje model vety alebo nie. Neprihliada sa k tomu, že symboly môžu mať interpretáciu ovplyvnenú tým, čo vlastne reprezentujú. Pre danú logickú vetu sa preto skúma všetkých 16 možných interpretácií a neprihliada sa k tomu, že zo sémantického hľadiska uvažovať platnosť symbolu vek_pod_17 a súčastne sa snažiť interpretovať symbol vek_pod_18 ako nepravdivý je nezmyselné.

Problémom je to, že symboly nemajú žiadnu vnútornú štruktúru, ktorá by vyjadrila, že nejaký symbol môže súvisieť s iným symbolom (napr. symboly vek_pod_17 a vek_pod_18 sa oba vzťahujú na vek nejakej osoby). Takáto informácia sa pri použití výrokovej logiky stráca, čo ochudobňuje jej vyjadrovacie možnosti pri použití ako modelovacieho nástroja.

Vnútorná štruktúra

Ak by však symboly mohli mať nejakú vnútornú štruktúru, uvedený fakt by mohol byť reprezentovaný napríklad v tvare logickej vety

$$skuska \land \neg(vek < 17) \land ((vek < 18) \to spolujazdec)$$

kde *skuska* a *spolujazdec* sú aj naďalej logickými symbolmi, avšak *vek* je celočíselná alebo reálna premenná. Výrazy $vek < 17$ a $vek < 18$ sú

ILUSTRAČNÝ PRÍKLAD

Úloha: *Kamaráti Fero, Ondro, Stano, Rudo a Jaro si chcú pripraviť niečo pod zub. Rúk je dostatok, problematickým je však vybavenie kuchyne. K dispozícii sú iba krájač (K), varič (V), mixér (M) a doska (D) na prípravu cesta. Dohodli sa na príprave niekoľkých jedál, pričom príprava každého z nich vyžaduje využívanie niektorých súčastí vybavenia kuchyne v presne stanovenom poradí a počas presne stanovenej doby, určenými receptom daného jedla:*

1. *krémová zeleninová polievka: Krájač 5 min, Varič 25 min, Mixér 5 min*
2. *šúľance so strúhankou: Varič 20 min, Doska 20 min, Varič 15 min*
3. *lekvárové pirohy: Doska 17 min, Varič 15 min*

kde napríklad príprava polievky najprv obsadí krájač na 5 časových jednotiek, po uvoľnení krájača obsadí varič a napokon po uvoľnení variča vyžaduje použitie mixéra. Pre správne vykonanie činností musí byť dodržané, že

- *činnosti každého receptu musia byť vykonávané v správnom poradí,*
- *každá činnosť musí trvať potrebnú dobu bez toho, aby bola prerušená,*
- *na každom kuse vybavenia môže v nejakej dobe byť vykonávaná činnosť iba jedného receptu.*

Úlohou je vytvoriť časové priradenie jednotlivých operácií na jednotlivé zariadenia umožňujúce aby všetky jedlá boli pripravené do hodiny a pol.
Reprezentácia: *Uvedený problém je reprezentovaný ako zložená logická veta, využívajúca atómy rozdielovej logiky, ktorá má tvar:*

$$
\begin{array}{ll}
 & (S_{1,K} \geq 0) \wedge (S_{2,V1} \geq 0) \wedge (S_{3,D} \geq 0) \\
\wedge & (S_{1,K} + 5 \leq S_{1,V}) \wedge (S_{1,V} + 25 \leq S_{1,M}) \wedge (S_{2,V1} + 20 \leq S_{2,D}) \\
\wedge & (S_{2,D} + 20 \leq S_{2,V2}) \wedge (S_{3,D} + 17 \leq S_{3,V}) \\
\wedge & (\, (S_{1,V} + 25 \leq S_{2,V1}) \vee (S_{2,V1} + 20 \leq S_{1,V}) \,) \\
\wedge & (\, (S_{1,V} + 25 \leq S_{2,V2}) \vee (S_{2,V2} + 15 \leq S_{1,V}) \,) \\
\wedge & (\, (S_{1,V} + 25 \leq S_{3,V}) \vee (S_{3,V} + 15 \leq S_{1,V}) \,) \\
\wedge & (\, (S_{2,V1} + 20 \leq S_{3,V}) \vee (S_{3,V} + 15 \leq S_{2,V1}) \,) \\
\wedge & (\, (S_{2,V2} + 15 \leq S_{3,V}) \vee (S_{3,V} + 15 \leq S_{2,V2}) \,) \\
\wedge & (\, (S_{2,D} + 20 \leq S_{3,D}) \vee (S_{3,D} + 17 \leq S_{2,D}) \,) \\
\wedge & (S_{1,M} + 5 \leq T_{max}) \wedge (S_{2,V2} + 15 \leq T_{max}) \wedge (S_{3,V} + 15 \leq T_{max}) \\
\wedge & (T_{max} \leq 90)
\end{array}
$$

Riešenie: *Pre celkový požadovaný čas je jedným z možných riešení voľnejší rozvrh* $S_{1,K} = 0$, $S_{1,V} = 5$, $S_{1,M} = 30$, $S_{2,V1} = 30$, $S_{2,D} = 50$, $S_{2,V2} = 70$, $S_{3,D} = 0$, *a* $S_{3,V} = 50$ *s celkovým časom 85 minút, zatiaľ čo iným z možných riešení je kompaktnejší rozvrh* $S_{1,K} = 0$, $S_{1,V} = 20$, $S_{1,M} = 45$, $S_{2,V1} = 0$, $S_{2,D} = 20$, $S_{2,V2} = 60$, $S_{3,D} = 0$ *a* $S_{3,V} = 45$ *s celkovým časom 75 minút.*

Ilustr. 4.1: Príklad pre reprezentáciu rozdielovou logikou naďalej interpretovateľné ako logicky pravdivé alebo nepravdivé (a teda je možné odvodiť pravdivosť vety ako celku), avšak táto interpretácia už nie je ľubovoľná ale záleží na

- hodnote premennej *vek*,
- interpretácii znaku $<$.

Modulová teória

Výroková logika bola doplnená *teóriou* $\mathcal{T}$ (často je označovaná ako *background* teória alebo *modulová* teória) – v uvedenom prípade $\mathcal{T}$ reprezentuje teóriu nerovností nad celými alebo reálnymi číslami. Nerovnosti $vek < 17$ a $vek < 18$ sú označované ako *atómy*, reprezentujúce výrazy zostavené zo stavebných blokov, poskytnutých teóriou $\mathcal{T}$. Interpretácia týchto atómov ako pravdivých alebo nepravdivých je daná vzhľadom na použitú teóriu $\mathcal{T}$.

Syntax

Uvedený prístup obohacuje reprezentáciu pomocou logických symbolov výrokovej logiky o používanie atómov v tom zmysle, ako sú definované v použitej modulovej teórii. Formálne je syntax vytváraných viet definovaná v Tab. 4.1.

$< \text{veta} >$	$::=$	$< \text{atomická veta} > \mid < \text{zložená veta} >$
$< \text{atomická veta} >$	$::=$	$\top \mid \perp \mid < \text{symbol} > \mid < \text{atóm} >$
$< \text{zložená veta} >$	$::=$	$< \text{unárny operátor} > < \text{veta} >$
	$\mid$	$< \text{veta} > < \text{binárny operátor} > < \text{veta} >$
	$\mid$	$(< \text{veta} >)$
$< \text{symbol} >$	$::=$	identifikátor
$< \text{unárny operátor} >$	$::=$	$\neg \mid \ldots$
$< \text{binárny operátor} >$	$::=$	$\vee \mid \wedge \mid \rightarrow \mid \leftrightarrow \mid \oplus \mid \uparrow \mid \downarrow \mid \ldots$

Tab. 4.1: Syntax výrazov modulových teórií

Syntakticky sa atóm dá rozložiť, má svoju vlastnú štruktúru. Táto štruktúra je však závislá od toho, aké prvky sú poskytované teóriou $\mathcal{T}$ a akým spôsobom ich táto teória umožňuje kombinovať.

Splniteľnosť

Pri hľadaní modelu sa na rozdiel od čisto výrokovej logiky nepožaduje všeobecná splniteľnosť ale splniteľnosť vzhľadom na nejakú teóriu $\mathcal{T}$, ktorá fixuje interpretáciu tých stavebných blokov, z ktorých sú vytvorené atómy. Takáto splniteľnosť sa označuje ako *$\mathcal{T}$-splniteľnosť*. Napríklad v prípade celočíselnej aritmetiky pri hľadaní modelu vety

$$\neg(vek < 17) \wedge \neg(vek < 18)$$

nie je záujem nájsť nejakú neštandardnú interpretáciu symbolu $<$, ktorá by zabezpečila platnosť uvedenej vety. Namiesto toho sa skúma, či daná veta môže byť pravdivá v interpretácii, v ktorej $<$ reprezentuje štandardné usporiadanie nad celými číslami.

KÓDOVANIE ATÓMOV

Vytváranie jednotlivých atómov pozostáva v zásade z dvoch po sebe idúcich krokov:

- identifikácia vhodnej modulovej teórie,
- mapovanie prvkov teórie na riešený problém.

Štruktúra problému stanovuje, aký typ informácií je potrebné reprezentovať – či je potrebné pracovať s numerickými hodnotami (a ak áno, aký typ hodnôt je potrebné použiť), s poliami, vektormi, funkciami, atď. Po identifikácie teórie ostáva už iba rozhodnúť, ktoré prvky problému budú reprezentované ktorými prvkami vybranej teórie.

Príklad 4.1 Použitie atómov nejakej modulovej teórie je možné ukázať na ilustračnom príklade zo strany 98. Keďže je potrebné reprezentovať časové priradenie jednotlivých operácií na jednotlivé zariadenia, tak základom reprezentácie budú numerické premenné označované $S_{x,y}$, kde index x reprezentuje číslo receptu a y reprezentuje operáciu na niektorom zo zariadení. Samotná premenná bude svojou hodnotou reprezentovať čas začiatku vykonávania operácie – tak napríklad $S_{1,K}$ reprezentuje čas začiatku krájania v prvom recepte zatiaľ čo $S_{2,V2}$ reprezentuje začiatok druhej fázy varenia v druhom recepte.

Žiadna operácia nemôže začať skôr ako začiatok samotného varenia. Pritom toto je potrebné definovať iba pre prvé operácie každého receptu, pretože ostatné operácie budú na tieto nadväzovať. Ak za začiatok varenia bude pre jednoduchosť zvolený čas 0, tak je potrebné definovať tri podmienky (ako tri atómy)

$$S_{1,K} \geq 0 \qquad S_{2,V1} \geq 0 \qquad S_{3,D} \geq 0$$

po jednej pre každý z receptov.

Poradie operácií v každom recepte je presne dané, žiadna operácia nemôže začať skôr než skončí predchádzajúca operácia daného receptu. Takýmto spôsobom je potrebné zabezpečiť súslednosť každej dvojice operácií, nasledujúcich za sebou. Keďže čas skončenia operácie sa dá vyjadriť ako súčet času jej začiatku a doby jej trvania, tak pre daný príklad je potrebné definovať päť podmienok

$$S_{1,K} + 5 \leq S_{1,V} \qquad \ldots \qquad S_{3,D} + 17 \leq S_{3,V}$$

začínajúcich od prvej dvojice operácií prvého receptu až po poslednú dvojicu operácií tretieho receptu.

Keďže na jednom zariadení môže byť vykonávaná v nejakom čase iba jedna operácia a až po jej skončení ďalšia operácia, tak pre každú dvojicu operácií z dvoch rôznych receptov, požadujúcich to isté zariadenie, je potrebné stanoviť, že jedna z nich môže začať až po ukončení druhej. Jediný problém je v tom, že nie je jasné, ktorá operácia z dvojice bude vykonaná skôr a ktorá neskôr – a teda je potrebné uvažovať každú z možností. Podmienka pre varenie v prvom recepte a prvú fázu varenia v druhom recepte bude mať tvar

$$S_{1,V} + 25 \leq S_{2,V1} \vee S_{2,V1} + 20 \leq S_{1,V}$$

kde prvá časť predpokladá vykonanie najprv operácie z prvého receptu a až po nej operácie z druhého receptu, zatiaľ čo druhá časť podmienky zohľadňuje poradie opačné. Keďže sa predpokladá použitie iba jedného poradia (avšak nevie sa ktorého), tak obe časti podmienky sú spojené disjunkciou. Takýchto podmienok je potrebných šesť – päť pre varič a jednu pre dosku na cesto.

Každý recept musí byť realizovaný maximálne do doby ukončenia varenia (čas T_{max}). Zabezpečiť sa to dá podmienkami, požadujúcimi aby posledná operácia každého receptu skončila skôr alebo presne v čase T_{max}. Dostaneme tri podmienky

$$S_{1,M} + 5 \leq T_{max} \qquad S_{2,V2} + 15 \leq T_{max} \qquad S_{3,V} + 15 \leq T_{max}$$

každú pre jeden z receptov.

No a keďže doba varenia je obmedzená, je to potrebné vyjadriť jednoduchou podmienkou

$$T_{max} \leq 90$$

zohľadňujúcou požadovanú maximálnu dobu prípravy.

Všetky definované podmienky musia byť splnené a preto musia byť spojené konjunkciou do výslednej reprezentácie v tvare jednej zloženej logickej vety.

•

POUŽÍVANÉ MODULOVÉ TEÓRIE

V predchádzajúcom bola ako modulová teória použitá teória čísel v spojení s operátormi + a ≤, definujúcimi skladanie čísel a ich vzájomné porovnávanie. Tým bol daný typický tvar použitých atómov. Súčasne štandardná sémantika sprístupnených operátorov poskytuje návod, ako interpretovať jednotlivé atómy z hľadiska ich pravdivosti alebo nepravdivosti. Pri použití inej modulovej teórie by atómy mohli byť skladané z iných stavebných elementov, mať iný typický tvar a ich interpretácia by mohla byť založená na inej sémentike.

Reprezentácia úlohy rozhoduje o tom, aká teória $\mathcal{T}$ by mala byť použitá v spojení s výrokovou logikou. Keďže však od použitej teórie závisí aj spôsob a efektivita procesu hľadania modelu, tak pri praktickom použití je typické používanie iba určitého malého počtu teórií. Najčastejšie používanými teóriami sú

- teória rovnosti s neinterpretovanými funkciami,
- lineárna aritmetika,
- aritmetika bitových vektorov,
- logika polí,
- smerníková logika.

V ďalšom sa zameriame na zrejme najčastejšie používanú *lineárnu aritmetiku* ako aj jeden jej fragment, často vystupujúci ako samostatná teória pod názvom *rozdielová logika*.

Rozdielová logika. Atómy používajú nerovnostné relačné operátory so štandardnou interpretáciou, definujúcou usporiadanie na množine čísel. Samotné atómy sú obmedzené na tvar

$$X - Y \bowtie c$$

kde $\bowtie$ reprezentuje operátor nerovnosti (teda niektorý z relačných operátorov $<$, $\leq$, $>$ a $\geq$). Uvedený tvar zahŕňa vždy dve premenné a jednu konštantu – jedná sa teda o vzájomné odlíšenie (rozdiel hodnôt) dvoch premenných[1].

Použitím tejto logiky je možné reprezentovať aj nasledujúce prípady, ktoré nie sú priamo v požadovanom tvare:

- $X - Y = c$ ako konjunkciu $(X - Y \leq c) \wedge (X - Y \geq c)$,
- $X < Y$ pomocou pridania nulovej konštanty $X - Y < 0$,
- $X < c$ ako $X - Z_o < c$ s pridaním špeciálnej premennej Z_0, ktorej hodnota musí byť vždy 0.

Pre negáciu (komplement) atómu platí $\neg(X - Y \bowtie c) = X - Y \neg \bowtie c$, kde $\neg \bowtie$ označuje zmenu, pri ktorej sa $<$ mení na $\geq$, $\leq$ na $>$, $>$ na $\leq$ a $\geq$ sa mení na $<$.

Premenné môžu byť definované nad oborom celých čísel (celčíselná rozdielová logika) alebo reálnych čísel (reálna rozdielová logika).

[1]Rozdielová logika sa preto niekedy označuje aj ako separačná logika.

Lineárna aritmetika. Atómy majú tvar lineárnych ohraničení v tvare rovností alebo nerovností (ostrých alebo neostrých). Sú teda ohraničené na niektorý z nasledujúcich tvarov

$$a_1X_1 + \ldots a_nX_n = b \qquad a_1X_1 + \ldots a_nX_n \bowtie b$$

kde $\bowtie$ opäť reprezentuje operátor nerovnosti, zatiaľ čo a_i a b reprezentujú číselné konštanty. Syntax atómov tejto teórie je uvedená v tab. 4.2.

< atóm >	::=	< výraz > < relácia > < výraz >
< výraz >	::=	< term > $\mid$ < výraz > < operátor > < výraz >
< term >	::=	< konštanta >
	$\mid$	< premenná >
	$\mid$	< konštanta > * < premenná >
< relácia >	::=	$= \mid < \mid \leq \mid > \mid \geq$
< operátor >	::=	+ $\mid$ -

Tab. 4.2: Syntax atómov lineárnej aritmetiky

Použitím tejto logiky je možné reprezentovať aj nasledujúce prípady iného tvaru

- $a_1X_1 \ldots a_nX_n < c_1Y_1 \ldots c_mY_m$ pomocou pridania nulovej konštanty $a_1X_1 \ldots a_nX_n - c_1Y_1 \ldots - c_mY_m < 0$,
- $a_1X_1 \ldots a_nX_n + b_1 < c_1X_1 \ldots c_nX_n + b_2$ pomocou zoskupenia konštánt $(a_1 - c_1)X_1 \ldots (a_n - c_n)X_n < b_2 - b_1$.

Pre negáciu (komplement) atómu platí rovnaká zmena ako v rozdielovej logike s výnimkou, keď negácia rovnosti sa mení na konjunkciu dvoch ostrých nerovností.

Niekedy sa používajú iba fragmenty tejto teórie – jedným takým fragmentom je aj rozdielová logika, obmedzujúca počet premenných (na dva), tvar ich koeficientov (na +1 a -1) a relačné operátory (na nerovnosti).

Hľadanie modelu

Pri hľadaní modelu vo výrokovej logike sa hľadala taká interpretácia výrokových symbolov, na základe ktorých veta ako celok bola interpretovaná ako pravdivá. Podobne sa teraz hľadá interpretácia výrokových symbolov a atómov použitej modulovaj teórie.

Ak takáto interpretácia nezohľadňuje štruktúru použitých atómov (jednoducho im priradzuje pravdivosť alebo nepravdivosť bez ohľadu na ich skutočnú štruktúru a vzájomné závislosti) a pritom vedie na pravdivosť vety, potom hovoríme o *výrokovom modele* danej vety, Tento výrokový model môže ale nemusí byť súčasne aj $\mathcal{T}$-splniteľný – v prípade že nie je, tak v konečnom dôsledku potom nie je modelom danej vety pri zohľadnení použitej modulovej teórie.

Príklad 4.2 Pre ilustráciu hľadania modelu uvažujme vetu pozostávajúcu z piatich literálov, vytvorených ako nad symbolmi tak aj nad atómami, tvaru

$$(P \vee Q) \wedge x - 3y \geq 0 \wedge \neg(x + 3y < 6) \wedge (\neg Q \vee y - x \geq 0)$$

kde P a Q sú výrokovými symbolmi a x a y sú reálnymi premennými.

Keďže nie je možné nájsť také dve reálne čísla, pre ktoré by podmienky $x - 3y \geq 0$ a $y - x \geq 0$ boli súčasne splnené, tak interpretácia

- $I = \{P^I = FALSE, Q^I = TRUE, (x - 3y \geq 0)^I = TRUE, (x + 3y < 6)^I = FALSE, (y - x \geq 0)^I = TRUE\}$ je výrokovým modelom danej vety, ktorý však nie je $\mathcal{T}$-splniteľný,
- $I = \{P^I = TRUE, Q^I = FALSE, (x - 3y \geq 0)^I = TRUE, (x + 3y < 6)^I = FALSE, (y - x \geq 0)^I = FALSE\}$ je výrokovým modelom vety a súčasne je aj $\mathcal{T}$-splniteľným,
- $I = \{P^I = FALSE, Q^I = TRUE, (x - 3y \geq 0)^I = TRUE, (x + 3y < 6)^I = FALSE, (y - x \geq 0)^I = FALSE\}$ nie je výrokovým modelom vety.

V konečnom dôsledku teda iba stredná interpretácia je modelom danej vety pri zohľadnení modulovej teórie.

•

ZÁKLADNÉ PRÍSTUPY K HĽADANIU

Efektívne hľadanie modelu by sa dalo dosiahnuť takým prispôsobením všeobecných inferenčných metód, aby zohľadňovali iba tie interpretácie, ktoré sú konzistentné s uvažovanou teóriou $\mathcal{T}$ – napríklad pomocou pridania axióm teórie $\mathcal{T}$. Toto však je možné iba pre niektoré teórie (napr. teória reálnych čísel nie je zachytiteľná konečnou množinou formulí). Preto schodnejšou alternatívou je použitie metód špecializovaných na použitú modulovú teóriu $\mathcal{T}$.

Spôsoby určenia splniteľnosti viet s ohľadom na použitú modulovú teóriu môžu byť rozdelené do dvoch základných prístupov:

- včasná kontrola,
- neskorá kontrola.

Tieto prístupy sa odlišujú tým, kedy sa zohľadňuje použitá modulová teória – pri včasnej kontrole proces začína touto teóriou, zatiaľ čo pri neskorej kontrole proces začína iným spôsobom a modulová teória sa zohľadňuje (kontroluje) v neskorších fázach.

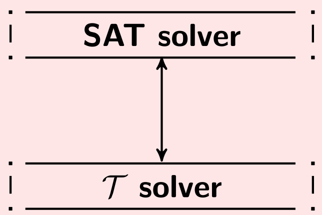
Včasná kontrola

Pri *včasnej kontrole* je veta transformovaná do tvaru výrokovej CNF (neobsahujúcej atómy iba symboly, z ktorých niektoré zastupujú atómy alebo aj samotnú teóriu), ktorej splniteľnosť môže byť kontrolovaná SAT solverom. Ak tento solver nájde výrokový model transformovanej vety, tento model bude súčasne $\mathcal{T}$-splniteľný.

Transformovaná veta je voči pôvodnej vete *ekvisplniteľná* – je splniteľná vtedy a iba vtedy, ak je splniteľná aj veta pôvodná. Transformácia je teda založená na použití sekvencie transformácií, ktoré zachovávajú splniteľnosť. Každá teória $\mathcal{T}$ vyžaduje ad-hoc transformácie, ktoré nie sú prenositeľné na inú teóriu. Problém nájdenia $\mathcal{T}$-splniteľného modelu je nahradený problémom nájdenia vhodnej transformácie do výrokovej CNF.

Neskorá kontrola

Alternatívou je *neskorá kontrola*, ktorá vyžaduje existenciu $\mathcal{T}$ solvera, schopného rozhodnúť o splniteľnosti konjunkcie atómov vytvorených podľa použitej teórie. Tento solver je kombinovaný so SAT solverom do štruktúry *SMT solvera* podľa obr. 4.1.



Obr. 4.1: Štruktúra SMT solvera

SAT solver slúži pre nachádzanie interpretácie, ktorá je výrokovým modelom skúmanej vety. Po nájdení takejto interpretácie je tá jej časť, ktorá sa týka atómov modulovej teórie, postúpená v tvare konjunkcie atómov na vstup $\mathcal{T}$ solvera, ktorý rozhodne o $\mathcal{T}$-splniteľnosti. V prípade, že konjunkcia atómov je označená za $\mathcal{T}$-nesplniteľnú, SAT solver pokračuje v hľadaní nového výrokového modelu. V opačnom prípade bola potvrdená vhodnosť výrokového modelu aj z hľadiska $\mathcal{T}$-splniteľnosti.

Častým reprezentantom uvedenej štruktúry je DPLL($\mathcal{T}$), kde SAT solver je založený na využití DPLL algoritmu (vrátane moderných doplnkov ako výberové heuristiky, analýza konfliktov a učenie sa nových klauzúl).

Príklad 4.3 Ak by v predchádzajúcom príklade, kde sa hľadal model vety

$$(P \vee Q) \wedge x - 3y \geq 0 \wedge \neg(x + 3y < 6) \wedge (\neg Q \vee y - x \geq 0)$$

SAT solver našiel interpretáciu $I = \{P^I = FALSE, Q^I = TRUE, (x-3y \geq 0)^I = TRUE, (x + 3y < 6)^I = FALSE, (y - x \geq 0)^I = TRUE\}$, tak na vstup $\mathcal{T}$ solvera postúpi konjunkcia atómov (pričom namiesto atómu, ktorý je interpretovaný ako nesplnený, je použitý jeho komplement)

$$x - 3y \geq 0 \wedge x + 3y \geq 6 \wedge y - x \geq 0$$

na kontrolu $\mathcal{T}$-splniteľnosti. •

NESKORÁ INKREMENTÁLNA KONTROLA

Pre zvýšenie efektívnosti hľadania môžu oba solvery spolupracovať *inkrementálnym* spôsobom. Nie je potrebné čakať, až SAT solver vygeneruje interpretáciu pre všetky použité symboly a atómy, ale už po vygenerovaní nejakej parciálnej interpretácie je možné zapojiť $\mathcal{T}$ solver pre testovanie atómov tejto parcialnej interpretácie. V prípade úspešného testu a ďalšieho rozšírenia parciálnej interpretácie o ďalší atóm, je $\mathcal{T}$ solver opäť aktivovaný, pričom v rámci testovania neopakuje toto testovanie odznovu ale spracuje v ňom iba tie testy, ktoré sa týkajú novo pridaného atómu.

$\mathcal{T}$ solver môže v prípade identifikácie nekonzistencie poskytnúť SAT solveru nielen informáciu o nekonzistencii skupiny atómov ale aj o príčine nekonzistencie – vo forme klauzuly, ktorej splnenosť vylučuje danú nekonzistenciu. Túto klauzulu môže SAT solver použiť pre zvýšenie svojej efektívnosti rovnakým spôsobom, ako používa klauzuly, ktoré sa naučil sám (napríklad solver založený na DPLL algoritme s doplneným učením konfliktných klauzúl).

URČOVANIE $\mathcal{T}$-SPLNITEĽNOSTI

V predchádzajúcom období boli objavené špecializované rozhodovacie procedúry pre celý rad teórií (či už kompletných alebo iba niektorých ich fragmentov), ktoré môžu byť použité v praktických aplikáciách v úlohe $\mathcal{T}$ solvera. Tieto procedúry umožňujú určovať splniteľnosť logických výrazov nad atómami modulových teórií vzhľadom na tieto modulové teórie. Preto pri praktickom použití je potrebné presne určiť, ktorá teória je použitá pre reprezentáciu úlohy.

Tieto rozhodovacie procedúry zvyčajne vyžadujú vstup v tvare konjunkcie atómov, bez použitia iných logických operátorov. V prípade vety všeobecného tvaru by preto najprv bolo potrebné transformovať vetu do tvaru DNF a potom postupne preskúmavať jednotlivé konjunkcie atómov z hľadiska ich $\mathcal{T}$-splniteľnosti. Alternatívou by bolo spájať atómy do konjunkcií podľa toho, či by sa vyskytli v konjunkcii po transformácii vety na DNF – avšak bez explicitného vykonávania tejto transformácie.

Pre kontrolu $\mathcal{T}$-splniteľnosti konjunkcie lineárnych nerovnostných ohraničení nad reálnymi premennými (konjunkcie atómov lineárnej aritmetiky s reláciou nerovnosti) je možné použiť *Fourier-Motzikinovu* metódu eliminácie premenných. Vstupom je m ohraničení nad n premennými x_1 až x_n tvaru FOURIER-MOTZIKINOVA ELIMINÁCIA PREMENNÝCH

$$\bigwedge_{i=1}^{m} \quad \sum_{j=1}^{n} a_{i,j} x_j \bowtie b_i$$

kde $\bowtie$ reprezentuje operátor nerovnosti (teda relačný operátor z množiny $\{<, \leq, >, \geq\}$) a koeficienty $a_{i,j}$ a b_i sú reálne konštanty.

Ak by sa medzi ohraničeniami vyskytli aj ohraničenia rovnosti (s relačným operátorom $=$), tak z každého takéhoto ohraničenia je možné vyjadriť jednu premennú v závislosti na ostatných premenných. Dosadením tejto závislosti za danú premennú do ostatných ohraničení sa dá daná premenná eliminovať a dané ohraničenie vypustiť. V prípade výskytu relačného operátora $\neq$ je možné dané ohraničenie nahradiť pomocou dvojice ohraničení ($>$ a $<$) a postup riešenia rozdeliť na dve vetvy – v každej z nich sa bude pracovať iba s jedným z týchto dvoch ohraničení. Týmto postupom je potrebné eliminovať všetky ohraničenia typu $=$ a $\neq$.

Princíp metódy je založený na výbere premennej a jej eliminácii, pričom ohraničenia danej premennej je potrebné zachovať v systéme ohraničení. Ak bola zvolená k-ta premenná x_k, tak potom i-te ohraničenie patrí do jednej z týchto troch kategórií:

- neutrálne ohraničenie,
- horné ohraničenie,
- dolné ohraničenie.

Neutrálne ohraničenie danú premennú nijako neohraničuje – premenná v danom ohraničení nevystupuje (a teda koeficient $a_{i,k}$ je nulový). Horné ohraničenie limituje maximálnu hodnotu, ktorú daná premenná môže nadobudnúť. Ohraničenie sa teda dá vyjadriť v tvare

$$x_k \leq \frac{1}{a_{i,k}} \left(b_i - \sum_{j=1}^{k-1} a_{i,j} x_j - \sum_{j=k+1}^{n} a_{i,j} x_j \right)$$

alebo v podobnom tvare s $\leq$ nahradeným ostrou nerovnosťou $<$. Zjednodušene sa ohraničenie dá vyjadriť ako $x_k \leq RHS_{i,k}$ alebo $x_k < RHS_{i,k}$, kde $RHS_{i,k}$ je pravá strana i-teho ohraničenia vyjadreného ako ohraničenie hodnoty k-tej premennej. Analogicky dolné ohraničenie limituje minimálnu

vstup: ohraničenia C nad reálnymi premennými
výstup: TRUE v prípade $\mathcal{T}$-splniteľnosti, FALSE inak

1. ZP = *zoznam_premennych(C)*
2. **foreach** $V \in ZP$ **do**
3. $\quad HO$ = *horne_ohranicenia(C,V)*
4. $\quad DO$ = *dolne_ohranicenia(C,V)*
5. $\quad NO$ = *nove_ohranicenia(DO,HO)*
6. $\quad C = C \setminus (HO \cup DO) \cup NO$
7. $\quad$ **if** *nesplnitelne(C)* **then return** FALSE
8. **return** TRUE

Alg. 4.1: Fourier-Motzikinova eliminácia premenných

hodnotu, ktorú daná premenná môže nadobudnúť. Takéto ohraničenie sa dá vyjadriť ako $x_k \geq RHS_{i,k}$ alebo $x_k > RHS_{i,k}$.

Alg. 4.1 reprezentuje postup vykonávania eliminácie premenných. Iteračným spôsobom sú postupne eliminované jednotlivé premenné (v každej iterácii jedna premenná) spolu s príslušnými úpravami množiny ohraničení až napokon ostanú iba také ohraničenia, o ktorých splniteľnosti alebo nesplniteľnosti je jednoduché rozhodnúť.

Pre nejakú premennú x_k sa najprv zistia jej horné a dolné ohraničenia (riadky 3 a 4). Eliminácia danej premennej sa odrazí na jednotlivých ohraničeniach v závislosti od kategórie týchto ohraničení:

- Neutrálne ohraničenia ostávajú nedotknuté.
- Ak premenná má iba horné ohraničenia, tak ich splnenie je jednoduché – stačí premennej dať dostatočne nízku hodnotu a budú splnené. Preto takéto ohraničenia je možné spolu s premennou odstrániť.
- Podobná situácia je ak premenná má iba dolné ohraničenia. Aj tieto je možné spolu s premennou odstrániť.
- Ak premenná má dolné aj horné ohraničenia, tak tieto sa nahradia novými ohraničeniami, v ktorých už daná premenná nevystupuje.

Ak predpokladáme, že i-te ohraničenie je dolným ohraničením premennej a j-te ohraničenie je horným ohraničením, tak musí platiť

$$\frac{x_k \geq RHS_{i,k} \qquad x_k \leq RHS_{j,k}}{RHS_{i,k} \leq RHS_{j,k}}$$

že dolné ohraničenie nemôže byť väčšie ako horné ohraničenie a táto podmienka musí byť pridaná k existujúcim ohraničeniam namiesto i-teho a j-teho ohraničenia, ktoré nahradí. Ak by sa v jednom z nahrádzaných ohraničení použila ostrá nerovnosť, tak potom ostrá nerovnosť bude aj v pridávanej podmienke.

Ak existujúce ohraničenia pre vypúšťanú premennú reprezentujú viac horných ohraničení alebo viac dolných ohraničení, tak potom je potrebné párovať každé dolné ohraničenie voči každému hornému ohraničeniu a pre každý takýto pár pridať príslušnú podmienku (riadok 5 algoritmu).

Záverečná úprava ohraničení (riadok 6) vzhľadom na elimináciu premennej x_k spočíva vo vypustení jej dolných a horných ohraničení a ich nahradení novo generovanými ohraničeniami.

Zložitosť eliminácie premenných

Ak by pri eliminácii premennej x_k napríklad pre túto premennú existovali dve dolné a štyri horné ohraničenia, tak potom týchto šesť ohraničení je nahradených ôsmimi novými ohraničeniami. Teda eliminácia premennej môže mať za následok nárast počtu ohraničení. V najhoršom možnom prípade, ak eliminovaná premenná sa vyskytuje vo všetkých m ohraničeniach, z ktorých polovica je dolným ohraničením a polovica horným ohraničením, tak týchto m ohraničení bude nahradených pomocou $m^2/4$ nových ohraničení – počet ohraničení narastá kvadraticky. Preto táto metóda je používaná iba pre riešenie úloh s menším počtom ohraničení a menším počtom premenných.

Aby sa čo najviac obmedzil rast počtu ohraničení, tak sa pre elimináciu v každej iterácii vyberá tá premenná, ktorá spôsobí pridanie najmenej nových ohraničení (teda súčin počtu horných ohraničení a počtu dolných ohraničení je najmenší).

Príklad 4.4 Pre ilustráciu metódy zisťovania splniteľnosti pomocou eliminácie premenných uvažujme konjunkciu nasledujúcich šiestich lineárnych ohraničení

$$\begin{array}{ccccccccc}
 & & & - & 3y & + & z & \leq & 0 \\
2w & + & x & & & & & \leq & 3 \\
 & - & x & + & y & & & \geq & 0 \\
 & & x & & & + & z & \geq & 6 \\
w & & & + & 2y & & & \leq & 2 \\
 & & x & - & 3y & & & \geq & 0
\end{array}$$

definovaných nad štyrmi premennými w, x, y a z. Ako prvú premennú eliminujme w. Táto premenná vystupuje v dvoch ohraničeniach, pričom obe sú hornými ohraničeniami hodnoty tejto premennej. Keďže žiadne ohraničenie nelimituje hodnotu tejto premennej zdola, tak ohraničenia sa jednoducho vypustia a získa sa skupina ohraničení

$$\begin{array}{rrrrrrrrr} & & & - & 3y & + & z & \leq & 0 \\ - & x & + & & y & & & \geq & 0 \\ & x & & & & + & z & \geq & 6 \\ & x & - & & 3y & & & \geq & 0 \end{array}$$

bez premennej w. Nech ďalšou eliminovanou premennou je z, ktorá sa vyskytuje v dvoch ohraničeniach – prvé ohraničenie je jej horným ohraničením $z \leq 3y$, zatiaľ čo tretie ohraničenie je jej dolným ohraničením $z \geq 6 - x$. Tieto dve ohraničenia budú nahradené novým ohraničením $x + 3y \geq 6$ a skupina ohraničení sa zmení na

$$\begin{array}{rrrrrr} & x & + & 3y & \geq & 6 \\ - & x & + & y & \geq & 0 \\ & x & - & 3y & \geq & 0 \end{array}$$

s prítomnosťou iba dvoch premenných. Premenná x sa vyskytuje vo všetkých troch ohraničeniach, pričom druhé ohraničenie je jej horným ohraničením $x \leq y$ a ostatné dve ohraničenia sú jej dolnými ohraničeniami $x \geq 6-3y$ a $x \geq 3y$. Existujúce tri ohraničenia budú nahradené dvomi novými

$$\begin{array}{rrr} 4y & \geq & 6 \\ 2y & \leq & 0 \end{array}$$

kde prvé vzniklo z $x \leq y$ a $x \geq 6 - 3y$, zatiaľ čo druhé pochádza z $x \leq y$ a $x \geq 3y$. Keďže teraz prvé ohraničenie je dolným ohraničením premennej y a druhé ohraničenie zase jej horným ohraničením, tak tieto dve ohraničenia možno nahradiť jedným ohraničením

$$1.5 \quad \leq \quad 0$$

ktoré je evidentne nesplnené – a teda ani pôvodná konjunkcia šiestich ohraničení nie je splniteľná.

•

Ak by ohraničenia po eliminácii všetkých premenných boli splniteľné, potom okrem potvrdenia splniteľnosti pôvodných ohraničení by bolo možné aj násť konkrétne hodnoty premenných, spĺňajúce ohraničenia – jednoducho by sa spätne išlo cez iterácie eliminácie premenných a v každej by sa tej premennej, ktorá v danej iterácii bola eliminovaná, priradila hodnota vyhovujúca horným a dolným ohraničeniam danej premennej.

Rozšírenie: Jednofázová simplexová metóda

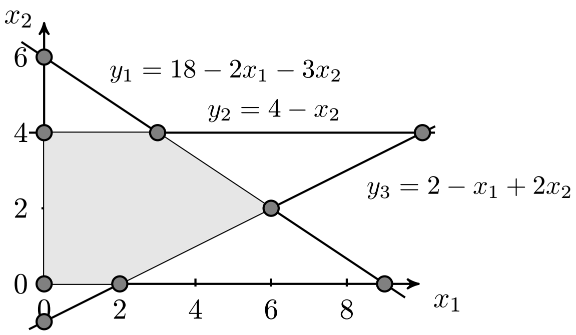
Simplexová metóda – princíp

Je to iteratívny postup pre hľadanie takého riešenia sústavy lineárnych ohraničení tvaru $\sum_{j=1}^{n} a_{i,j}x_j \leq b_i$ pre $b_i \geq 0$, ktoré maximalizuje kritérium v tvare lineárnej kombinácie hodnôt premenných $c_1x_1 + \ldots + c_nx_n$ nad sadou n nezáporných premenných x_1 až x_n.

Hodnoty premenných musia vyhovovať m ohraničeniam, ktoré je možné transformovať do kanonického tvaru, ktorý pridáva m nezáporných prídavných premenných y_1 až y_m:

$$\bigwedge_{i=1}^{m} \quad y_i = b_i - \sum_{j=1}^{n} a_{i,j}x_j \qquad b_i \geq 0, y_i \geq 0, x_j \geq 0$$

Takto zadané ohraničenia vytvárajú konvexnú oblasť možných riešení. Metóda delí $m+n$ premenných na m *základných* a n *nezákladných premenných* – hodnoty základných premenných sa určujú podľa hodnôt nezákladných premenných, ktorých hodnoty je možné voliť. *Základné riešenie* vznikne tak, že všetky nezákladné premenné sa zvolia nulové (a hodnoty základných premenných sa dopočítajú).



Obrázok zobrazuje príklad sústavy troch ohraničení nad dvomi premennými x_1 a x_2, ku ktorým kanonický tvar pridal ďalšie tri premenné. Vyplnená konvexná plocha reprezentuje oblasť možných riešení. Všetky základné riešenia sú zobrazené v tvare bodov – nie všetky sú súčasne možnými riešeniami spĺňajúcimi všetky podmienky.

Metóda neprehľadáva celú oblasť možných riešení ale iteračným spôsobom skúma iba tie základné riešenia, ktoré sú možnými riešeniami, pričom sa snaží zlepšovať hodnotu kritéria. Začína v základnom riešení, v ktorom základnými premennými sú prídavné premenné (teda $x_1 = \ldots = x_n = 0$). Medzi jednotlivými základnými riešeniami prechádza tak, že vždy vymieňa navzájom jednu základnú a jednu nezákladnú premennú.

Rozšírenie: Jednofázová simplexová metóda (pokr.)

SIMPLEXOVÁ METÓDA – PRÍKLAD

Pre ukážku postupu uvažujme dve premenné $x_1 \geq 0$ a $x_2 \geq 0$, maximalizačné kritérium $x_1 + 2x_2$ a tri rovnice

$$\begin{array}{lcll} y_1 & = & 18 - 2x_1 - 3x_2 & \quad y_1 \geq 0 \\ y_2 & = & 4 - x_2 & \quad y_2 \geq 0 \\ y_3 & = & 2 - x_1 + 2x_2 & \quad y_3 \geq 0 \end{array}$$

Základnými premennými sú y_1, y_2 a y_3, nezákladnými premennými x_1 a x_2. Základným riešením je $x_1 = 0$, $x_2 = 0$, $y_1 = 18$, $y_2 = 4$ a $y_3 = 2$ s hodnotou kritéria 0.

Keďže kritérium je $x_1 + 2x_2$, je ho možné zvýšiť pomocou zvýšenia hodnoty premennej x_1 – nezákladné premenné teda budú $x_1 \geq 0$ a $x_2 = 0$, čo po dosadení do rovníc dá horné ohraničenia pre x_1

$$\begin{array}{lcll} y_1 & = & 18 - 2x_1 & \quad y_1 \geq 0 \rightarrow x_1 \leq 9 \\ y_2 & = & 4 & \quad y_2 \geq 0 \rightarrow x_1 \leq \infty \\ y_3 & = & 2 - x_1 & \quad y_3 \geq 0 \rightarrow x_1 \leq 2 \end{array}$$

a teda premennú x_1 možno zvýšiť najviac na hodnotu 2, následkom čoho bude $y_3 = 0$. Premenná x_1 tak prestáva byť nezákladnou premennou (lebo už nie je nulová) – namiesto nej sa môže stať základnou premennou y_3 (lebo už je nulová). Je potrebné teda vymeniť navzájom premenné x_1 a y_3. Z poslednej rovnice (jedinej v ktorej sa vyskytujú spolu x_1 aj y_3) sa vyjadrí nová nezákladná premenná v tvare $x_1 = \ldots$ a zároveň sa urobí dosadenie za x_1 aj v ostatných rovniciach. Výsledkom bude nová podoba ohraničení

$$\begin{array}{lcll} y_1 & = & 14 - 7x_2 + 2y_3 & \quad y_1 \geq 0 \\ y_2 & = & 4 - x_2 & \quad y_2 \geq 0 \\ x_1 & = & 2 + 2x_2 - y_3 & \quad x_1 \geq 0 \end{array}$$

a maximalizačné kritérium po dosadení x_1 sa zmení na $2 + 4x_2 - y_3$.

Po prvej iterácii základnými premennými sú y_1, y_2 a x_1, nezákladnými premennými x_2 a y_3. Základným riešením je $x_2 = 0$, $y_3 = 0$, $y_1 = 14$, $y_2 = 4$ a $x_1 = 2$ so zvýšenou hodnotou kritéria 2.

Hodnotu kritéria je ďalej možné zvýšiť pomocou zvýšenia hodnoty x_2 (y_3 sa nedá znížiť kvôli $y_3 \geq 0$). Opakovaním postupu by po druhej iterácii základnými premennými boli y_2, x_2 a x_1, nezákladnými premennými y_1 a y_3. Novým základným riešením by bolo $y_1 = 0$, $y_3 = 0$, $y_2 = 2$, $x_1 = 6$ a $x_2 = 2$ s hodnotou kritéria zvýšenou na 10.

V ďalšej iterácii je možné ďalej zvyšovať hodnotu kritéria a opäť z jednej nezákladnej premennej urobiť základnú a naopak.

Inou možnosťou pre zistenie $\mathcal{T}$-splniteľnosti konjunkcie lineárnych nerovnostných ohraničení nad reálnymi premennými (konjunkcie atómov lineárnej aritmetiky s reláciou nerovnosti) je použitie *simplexovej metódy.* Základné použitie tejto metódy je pre hľadanie optimálneho riešenia, ktoré maximalizuje hodnotu lineárneho maximalizačného kritéria a zároveň spĺňa požiadavky všetkých ohraničení. Je však možné túto metódu určitým spôsobom použiť aj pre zistenie splniteľnosti ohraničení.

Simplexová metóda pre splniteľnosť

V ideálnom prípade by atómy mali tvar lineárnych neostrých nerovností nad n premennými x_1 až x_n tvaru

$$\bigwedge_{i=1}^{m} \sum_{j=1}^{n} a_{i,j} x_j \leq b_i$$

pričom pre všetky premenné platí podmienka nezápornosti $x_j \geq 0$ rovnako ako aj pre všetky koeficienty na pravej strane nerovníc $b_i \geq 0$. Pri takomto tvare ohraničenia sú splniteľné – minimálne prostredníctvom triviálneho priradenia $x_1 = 0, \ldots, x_n = 0$. Preto je žiadúce existujúce ohraničenia transformovať na takýto tvar – ak by sa to podarilo, bola by istota že sú splniteľné (pomocou triviálneho nulového priradenia).

Pri transformácii danej množiny lineárnych ohraničení na uvedený tvar je možné pre niektoré tvary použiť

- $LHS_i \geq b_i$ je možné násobením hodnotou -1 zmeniť na požadované $-LHS_i \leq -b_i$,
- $LHS_i = b_i$ je možné nahradiť dvomi ohraničeniami $LHS_i \geq b_i$ a $LHS_i \leq b_i$ (alebo vyjadriť hodnotu jednej premennej pomocou ostatných premenných a dosadením do zostávajúcich ohraničení túto premennú eliminovať).

V prípade iného definičného oboru premenných ako $x_i \geq 0$ je možné využiť transformácie

- $c_i \leq x_i$ je možné nahradením premennej x_i za novú premennú $x_i^{'}$ (dosadením $x_i = c_i + x_i^{'}$) zmeniť na požadované $x_i^{'} \geq 0$,
- $c_i \geq x_i$ je možné nahradením premennej x_i za novú premennú $x_i^{'}$ (dosadením $x_i = c_i - x_i^{'}$) zmeniť na požadované $x_i^{'} \geq 0$,
- $-\infty \leq x_i \leq \infty$ je možné nahradením premennej x_i za dvojicu premenných x_i^1 a x_i^2 (dosadením $x_i = x_i^1 - x_i^2$) zmeniť na požadované $x_i^1 \geq 0$ a $x_i^1 \geq 0$.

Jediný problém nastáva v prípade, ak po úprave na požadovaný tvar nie je dodržaná podmienka nezápornosti konštánt b_i. Aj vtedy môže byť skupina atómov splniteľná, avšak triviálne nulové priradenie premenných už nemusí splňať všetky ohraničenia.

V tomto prípade je potrebné zaviesť špeciálnu premennú $z \geq 0$ a pridať ju ku každému z lineárnych ohraničení

$$\bigwedge_{i=1}^{m} \quad \sum_{j=1}^{n} a_{i,j} x_j - z \leq b_i$$

kde dostatočne veľká hodnota premennej z spolu s nulovými hodnotami premenných x_1 až x_n zabezpečia splnenie všetkých ohraničení. Takýto systém nerovníc je možné previesť pomocou pridania prídavných premenných y_1 až y_m na kanonický systém rovníc

$$\bigwedge_{i=1}^{m} \quad y_i = b_i - z - \sum_{j=1}^{n} a_{i,j} x_j$$

pričom je potrebné maximalizovať kritérium $-z$. Takýto systém rovníc je vždy splniteľný (lebo vhodnou veľkou hodnotou z ho takým možno urobiť). Maximalizácia daného kritéria má za cieľ hľadať čo najmenšiu hodnotu premennej z, pri ktorej je systém rovníc ešte splniteľný. Ak sa pri tej maximalizácii zistí, že je to možné aj pre $z = 0$, tak to znamená, že splniteľným bol aj pôvodný súbor nerovníc ešte pred pridaním tejto premennej.

Keďže daný systém rovníc je v kanonickom tvare požadovanom jednofázovou simplexovou metódou, pre optimalizáciu kritéria je možné použiť túto metódu. Jediný problém je v tom, že pri základnom riešení hodnoty základných premenných nie sú všetky nezáporné. Preto v rámci prvej iterácie je potrebné vymeniť nezákladnú premennú z a tú základnú premennú y_i, ktorá má najmenšiu hodnotu b_i.

PRIAME KÓDOVANIE PRE ROZDIELOVÚ LOGIKU

Pre ilustráciu prístupu z kategórie včasnej kontroly uveďme metódu *priameho kódovania* pre určovanie splniteľnosti logických výrazov nad atómami rozdielovej logiky (táto metóda nie je obmedzená iba na konjunkciu atómov tejto logiky). Vstupom je veta nad atómami, pričom sa tieto atómy uvažujú v tvare

$$X \geq Y + c$$

jednoducho získateľnom z tvarov $X - Y \geq c$ alebo $Y - X \leq -c$ používaných rozdielovou logikou.

Táto veta je transformovaná na ekvisplniteľnú vetu nad symbolmi, zastupujúcimi ako atómy pôvodnej vety tak aj atómy, ktoré sa síce v pôvodnej

vstup: veta F_a nad atómami rozdielovej logiky
výstup: veta F_s nad symbolmi

F_{skel} = *generovanie_skeletonu*(F_a)
$F_{ohr} = \top$
ZP = *zoznam_premennych*(F_a)
ZA = *zoznam_atomov*(F_a)
foreach $V \in ZP$ **do**
    O = *generuj_ohranicenia*(V,ZA)
    $ZA = ZA \wedge$ *nove_atomy*(O)
    ZA = *vypusti_premennu*(V,ZA)
    $F_{ohr} = F_{ohr} \wedge$ *generovanie_skeletonu*(O)
$F_s = F_{skel} \wedge F_{ohr}$

Alg. 4.2: Priame kódovanie viet rozdielovej logiky

vete nevyskytovali, avšak vyplývajú z vlastností použitých relačných operátorov. Alg. 4.2 reprezentuje postup vykonávania tejto transformácie.

Pod generovaním skeletonu sa rozumie náhrada každého atómu spracovávanej vety zástupným symbolom, ktorý bude daný atóm reprezentovať. Pre jednoduchosť atóm $X \geq Y + c$ nahradíme logickým symbolom $s^c_{X,Y}$, čo zachová väzbu medzi symbolom a ním reprezentovaným atómom – symbol $s^c_{X,Y}$ bude teda reprezentovať atóm, ktorý požaduje aby hodnota premennej X (prvý dolný index) bola väčšia od hodnoty premennej Y (druhý spodný index) aspoň o hodnotu c (horný index). Tento symbol bude interpretovaný ako pravdivý, ak hodnoty oboch príslušných číselných premenných spĺňajú ohraničenie, dané atómom.

Ak sa v spracovávanej vete vyskytne atóm, ktorému ešte nebol vytvorený symbol, tak sa vytvorí nový symbol. Ak sa v spracovávanej vete vyskytne nejaký atóm viackrát, potom aj jeho zástupný symbol bude použitý viackrát.

Príklad 4.5 Tak napríklad veta nad atómami rozdielovej logiky tvaru

$$((X \geq Y + 3) \vee (Y \geq Z + 1)) \wedge (\neg(X \geq Y + 3) \vee \neg(Y \geq Z + 1))$$

bude nahradená vetou

$$\left(s^3_{X,Y} \vee s^1_{Y,Z}\right) \wedge \left(\neg s^3_{X,Y} \vee \neg s^1_{Y,Z}\right)$$

pričom boli vygenerované dva symboly.

V prvom kroku (krok 1) je teda každý atóm zamenený za symbol a výsledkom je symbolová verzia vstupnej vety F_a, kopírujúca jej štruktúru. Táto veta je označená ako F_{skel}.

V ďalšej časti sa algoritmus inšpiroval Fourier-Motzikinovou elimináciou premenných – postupne zo zoznamu premenných ZP bude eliminovať jednotlivé číselné premenné. Tieto premenné bude uvažovať v určitom vytvorenom poradí (krok 3). Zároveň bude zo zoznamu atómov ZA eliminovať tie atómy (ohraničenia), ktoré súvisia s eliminovanými premennými. Iniciálne je ZA vytvorený ako zoznam všetkých tých atómov, ktoré sa nachádzajú v štruktúre vstupnej vety (krok 4).

Pre každú premennú sa bude skúmať, či tie atómy, ktoré obsahujú danú premennú, je možné jednoducho odstrániť alebo či je potrebné ich nahradiť novými atómami (ohraničeniami), projektujúcimi vplyv tejto premennej na ostatné premenné. Ak boli nájdené takéto nové ohraničenia (krok 6), tak sa skúma, či tieto ohraničenia obsahujú atómy, ktoré ešte nie sú v zozname atómov ZA. Ak áno, tak sa nové atómy pridajú k aktuálnej podobe ZA (krok 7). Po takomto spracovaní sú z aktuálneho zoznamu atómov ZA vypustené všetky atómy, obsahujúce danú premennú (krok 8). Nové ohraničenia sú vyjadrené aj v symbolickej podobe s prípadným zavedením nových symbolov a v takejto podobe sú akumulované vo vete F_{ohr} (krok 9 algoritmu).

Po eliminácii všetkých premenných je výsledná veta F_s nad symbolmi (krok 10) vytvorená zlúčením vety F_{skel} (symbolová reprezentácia pôvodnej vety nad atómami rozdielovej logiky) a vety F_{ohr} (symbolová reprezentácia dodatočných ohraničení, nahrádzajúcich pôsobenie jednotlivých premenných).

Kľúčovým krokom je generovanie nových ohraničení, reflektujúcich vlastnosti použitej modulovej teórie, ktoré by sa stratili náhradou atómov pomocou symbolov – preto je ich nutné explicitne vyjadriť.

Jednou takouto vlastnosťou je tranzitívnosť operátorov reprezentujúcich nerovnosť. Toto sa dá vyjadriť ako

$$\frac{X \geq Y + c_1 \qquad Y \geq Z + c_2}{X \geq Z + (c_1 + c_2)}$$

čo je analógia prístupu Fourier-Motzikinovej eliminácie pri generovaní nového ohraničenia v prípade, ak eliminovaná premenná má súčasne dolné aj horné ohraničenie. Ak by bola premenná Y eliminovaná a oba atómy, v ktorých sa vyskytuje boli vypustené, tak by sa stratila informácia o vzťahu medzi premennými X a Z. Preto je potrebné túto informáciu zachovať. Ak atóm $X \geq Z + (c_1 + c_2)$ ešte nie je v zozname známych atómov ZA, tak sa doň pridá. A symbolová podoba ohraničenia

$$X \geq Y + c_1 \wedge Y \geq Z + c_2 \rightarrow X \geq Z + (c_1 + c_2)$$

sa pridá k ohraničeniam, ktoré ostávajú explicitne vyjadrené v F_{ohr}.

Inou vlastnosťou je neprotirečivosť v situácii, keď vzťah medzi dvomi premennými je dvojstranný (teda hodnota jednej premennej je ohraničená hodnotou druhej premennej a súčasne hodnota druhej premennej je ohraničená hodnotou prvej premennej). Pri výskyte dvoch atómov

$$X \geq Y + c_1 \qquad Y \geq X + c_2$$

by podľa predchádzajúceho pravidla po eliminácii premennej X vzniklo nové ohraničenie

$$X \geq Y + c_1 \wedge Y \geq X + c_2 \rightarrow c_1 + c_2 \leq 0$$

V prípade platnosti $c_1 + c_2 \leq 0$ by sa dané ohraničenie zmenilo na

$$X \geq Y + c_1 \wedge Y \geq X + c_2 \rightarrow TRUE$$

čo je vždy platná veta (tautológia) a netreba ju preto uvažovať. Avšak v prípade neplatnosti $c_1 + c_2 \leq 0$ si pôvodné atómy protirečia – dané nové ohraničenia by sa zmenilo na

$$X \geq Y + c_1 \wedge Y \geq X + c_2 \rightarrow FALSE$$

čo je veta platná iba vtedy, ak ľavá strana implikácie je nesplnená. Ak je teda súčet oboch konštánt kladný, potom nie je možné splniť obe podmienky naraz. Explicitne sa to vyjadrí pridaním symbolovej podoby ohraničenia

$$\neg(X \geq Y + c_1) \vee \neg(Y \geq X + c_2)$$

k explicitne vyjadreným ohraničeniam v F_{ohr}. Keďže v rámci nového ohraničenia nebol generovaný žiadny nový atóm, nie je potrebné nič pridať do zoznamu atómov ZA.

Rovnako je vhodné uvažovať vlastnosť viacnásobnosti v situácii, keď vzťah medzi dvoma premennými je vyjadrený viacnásobne (teda hodnota premennej je obmedzovaná hodnotou inej premennej viackrát). Pri výskyte dvoch atómov

$$X \geq Y + c_1 \qquad X \geq Y + c_2$$

v prípade $c_1 \neq c_2$ jedna z podmienok (tá s väčšou konštantou) implikuje podmienku druhú a teda z jej splnenia vyplýva splnenie podmienky s menšou konštantou. Ak teda platí $c_1 > c_2$, potom je potrebné symbolovú podobu ohraničenia

$$X \geq Y + c_1 \rightarrow X \geq Y + c_2$$

pridať k ostatným explicitne reprezentovaným ohraničeniam v F_{ohr}. Do zoznamu ZA sa nič nepridáva, pretože podmienka negenerovala žiadny nový atóm.

Príklad 4.6 Pre ilustráciu použitia priameho kódovania rozdielovej logiky uvažujme vstupnú vetu F_a s atómami nad tromi číselnými premennými a, b a c v tvare

$$\begin{aligned} c \geq a+1 \ \ \wedge & \ \ (a \geq b+3 \vee b \geq a+5) \ \wedge \ (c \geq b+3 \vee b \geq c+4) \\ \wedge & \ \ (c \geq b+3 \vee c \geq b-6) \ \wedge \ (b \geq c-5 \vee b \geq c+4) \\ \wedge & \ \ (b \geq c-5 \vee c \geq b-6) \end{aligned}$$

Štruktúra tejto vety bude zachytená v F_{skel} pri použití siedmich nových symbolov

$$\begin{aligned} F_{skel} \ = \ & s^1_{c,a} \wedge (s^3_{a,b} \vee s^5_{b,a}) \wedge \\ & (s^3_{c,b} \vee s^4_{b,c}) \wedge (s^3_{c,b} \vee s^{-6}_{c,b}) \wedge (s^{-5}_{b,c} \vee s^4_{b,c}) \wedge (s^{-5}_{b,c} \vee s^{-6}_{c,b}) \end{aligned}$$

zoznam premenných je $ZP = \{a, b, c\}$ a zoznam atómov ZA obsahuje sedem členov

$$a \geq b+3, b \geq a+5, b \geq c+4, b \geq c-5, c \geq a+1, c \geq b+3, c \geq b-6$$

Ako prvá bude eliminovaná premenná a, ktorej v zozname atómov zodpovedajú tri atómy $a \geq b+3$, $b \geq a+5$ a $c \geq a+1$. S touto premenou sa viažu dve nové ohraničenia a preto po jej eliminácii sa veta F_{ohr} zmení na

$$F_{ohr} \ = \ (s^3_{a,b} \wedge s^1_{c,a} \rightarrow s^4_{c,b}) \wedge (\neg s^3_{a,b} \vee \neg s^5_{b,a})$$

čo je ekvivalentné

$$F_{ohr} \ = \ (\neg s^3_{a,b} \vee \neg s^1_{c,a} \vee s^4_{c,b}) \wedge (\neg s^3_{a,b} \vee \neg s^5_{b,a})$$

pričom bol generovaný nový symbol $s^4_{c,b}$. Zoznam atómov bude mať tvar

$$ZA \ = \ b \geq c+4, b \geq c-5, c \geq b+3, c \geq b-6, c \geq b+4$$

s piatimi atómami.

Druhou eliminovanou premennou je premenná b, ktorá sa vyskytuje vo všetkých ostávajúcich atómoch v zozname atómov. Z hľadiska neprotirečivosti pribudnú dve ohraničenia do vety F_{ohr}

$$F_{ohr} \quad = \quad F_{ohr} \wedge (\neg s^4_{b,c} \vee \neg s^3_{c,b}) \wedge (\neg s^4_{b,c} \vee \neg s^4_{c,b})$$

bez generovania nových symbolov a rozšírenia zoznamu atómov. Z hľadiska viacnásobnosti pribudnú do F_{ohr} štyri nové ohraničenia

$$F_{ohr} \quad = \quad F_{ohr} \wedge (\neg s^4_{b,c} \vee s^{-5}_{b,c}) \wedge (\neg s^4_{c,b} \vee s^3_{c,b}) \wedge (\neg s^4_{c,b} \vee s^{-6}_{c,b}) \wedge (\neg s^3_{c,b} \vee s^{-6}_{c,b})$$

opäť bez generovania nových symbolov a rozšírenia zoznamu atómov. •

Pre určenie splniteľnosti vety $F_s = F_{skel} \wedge F_{ohr}$ je možné použiť štandardný SAT solver (pre hľadanie interpretácie ôsmich symbolov pri štrnástich klauzulách) – výsledkom je nájdenie viacerých modelov, čo znamená že F_s je splniteľná a vďaka ekvisplniteľnosti je zároveň aj pôvodná veta F_a $\mathcal{T}$-splniteľná.

Ak by bolo potrebné nielen zistiť $\mathcal{T}$-splniteľnosť vety F_a ale aj nájsť nejaké riešenie s hodnotami číselných premenných, je možné zobrať niektorý z výrokových modelov, nájdených SAT solverom, s následným hľadaním vhodných hodnôt premenných, zachovávajúcich platnosť výrokového modelu.

NÁJDENIE HODNÔT PREMENNÝCH

Je potrebné teda nájsť také hodnoty jednotlivých premenných, pri ktorých všetky atómy vybraného výrokového modelu musia byť pre dané riešenie splnené. Samotnému hľadaniu môže predchádzat fáza zjednodušovania – atómy, ktoré reprezentujú redundantné ohraničenia, sú eliminované. Napríklad dvojica premenných môže byť navzájom ohraničovaná viacerými ohraničeniami. Kvôli redukcii ich počtu sa ponechajú iba najprísnejšie – to sú také, ktorých splnenie zabezpečí aj splnenie ostatných ohraničení viazaných na danú dvojicu premenných.

Príklad 4.7 V predchádzajúcom príklade jeden z modelov bol

$$\neg s^3_{a,b} \wedge s^5_{b,a} \wedge s^3_{c,b} \wedge \neg s^4_{b,c} \wedge s^1_{c,a} \wedge s^{-5}_{b,c} \wedge s^{-6}_{c,b} \wedge \neg s^4_{c,b}$$

a teda interpretoval ako splnené tieto atómy

$$a < b+3, b \geq a+5, c \geq b+3, b < c+4, c \geq a+1, b \geq c-5, c \geq b-6, c < b+4$$

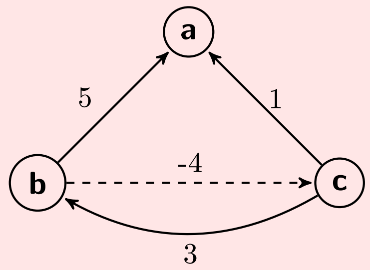
pričom tie atómy, ktorých zástupný symbol bol interpretovaný ako nesplnený, boli nahradené svojimi komplementmi.

Premenné a a b sú ohraničované pomocou $a < b+3$ a $b \geq a+5$. Z tejto dvojice ohraničení stačí uvažovať iba $b \geq a+5$. Kontrolou všetkých dvojíc získame redukovanú množinu ohraničení

$$b \geq a+5, c \geq b+3, c \geq a+1, c < b+4$$

obsahujúcu iba štyri ohraničenia. •

Tieto ohraničenia je možné znázorniť v tvare *grafu ohraničení*. Jednotlivé premenné sú reprezentované ako uzly a ohraničenia sú znázornené ako orientované hrany, kde plná hrana reprezentuje $\geq$ a čiarkovaná zase $>$ typ nerovnosti. Ukážka takéhoto grafu, zodpovedajúceho štyrom ohraničeniam predchádzajúceho príkladu je na obr. 4.2.



Obr. 4.2: Graf ohraničení

Aj z grafu vidno, že existuje riešenie – v grafe nie je žiadna kladná slučka (pre ktorú by súčet váh jej hrán bol kladný). Riešenie je možné získať postupným priradzovaním hodnôt jednotlivým premenným.

Príklad 4.8 Uvažujme priradenie hodnôt premenným, vyhovujúce grafu ohraničení podľa obr. 4.2. Keďže medzi premennými nebola špeciálna premenná Z_0, žiadna premenná zatiaľ nemá stanovenú žiadnu hodnotu. Môžme teda začať trebárs premennou a a jej priradením $a = 0$.

Z grafu je zrejmé, že pri uvážení priamych aj nepriamych (tranzitívnych cez iné uzly) ciest medzi jednotlivými uzlami a výbere najdlhšej cesty platí pre hodnoty ostávajúcich premenných $b \geq a+5 = 5$ a $c \geq a+8 = 8$.

Premennej b možno priradiť trebárs $b = 6$. Pre poslednú premennú podľa grafu musí platiť $c \geq b+3 = 9$ a súčasne aj $c-4 < b = 6$ (a teda $c < 10$). Poslednej premenej možno prisúdiť $c = 9$. •

Iným možným spôsobom hľadania konkrétneho priradenia hodnôt jednotlivým číselným premenným je využitie faktu, že výrokové modely vety

F_s sú konjunkciami atómov. Je teda možné použiť niektorú z metód riešenia konjunkcie lineárnych ohraničení (napríklad Fourier-Motzikinovu elimináciu premenných) nie v zmysle stanovenia splniteľnosti ale hľadania priradenia hodnôt jednotlivým premenným.

Alg. 4.2 je možné použiť nielen nielen pre rozdielovú logiku ale aj pre lineárnu aritmetiku nad reálnymi premennými, kde jednotlivé atómy sú v tvare PRIAME KÓDOVANIE PRE LINEÁRNU ARITMETIKU

$$\sum_{j=1}^{n} a_{i,j} x_j \bowtie b_i$$

Všeobecne vyjadrenú vetu nad takto definovanými atómami je potrebné normalizovať pomocou troch krokov:

- rovnosti nahradiť konjunkciou nerovností,
- transformovať vetu do tvaru NNF (negatívnej normálnej formy), kde negovanými môžu byť iba atómy,
- nahradiť negáciu zmenou relačných operátorov.

Podobne ako pri rozdielovej logike, štruktúra výslednej vety sa zachytí v štruktúre skeletonu F_{skel} (riadok 1 algoritmu). Závislosti medzi dolnými a hornými hranicami jednotlivých premenných budú zachytené v F_{ohr} (riadky 2 a 9). Tieto závislosti sú vyhľadávané počas iteračnej eliminácie premenných, pričom odvodzovacie pravidlo

$$\frac{x_k \geq RHS_{i,k} \qquad x_k \leq RHS_{j,k}}{RHS_{i,k} \leq RHS_{j,k}}$$

je použité pre generovanie nových ohraničení. Ak teda eliminovaná premenná má dolnú aj hornú hranicu, potom ak symbol s_i reprezentuje podmienku $x_k \geq RHS_{i,k}$ a symbol s_j zase podmienku $x_k \leq RHS_{j,k}$, tak do F_{ohr} sa pridá

$$s_i \wedge s_j \rightarrow s_k$$

kde nový symbol s_k reprezentuje novú podmienku $RHS_{i,k} \leq RHS_{j,k}$. Ak však táto nová podmienka je nesplnená, potom do F_{ohr} sa pridá

$$s_i \wedge s_j \rightarrow FALSE$$

bez vytvárania nového symbolu s_k.

Príklad 4.9 Pre ilustráciu priameho kódovania lineárnej aritmetiky uvažujme vstupnú vetu s atómami nad tromi číselnými premennými a, b a c v tvare

$$-a + 2c \leq -3 \wedge \neg(2a - 2b > -1 \wedge b - 2c \geq -1)$$

ktorú je možné normalizovať do tvaru

$$-a + 2c \leq -3 \wedge (2a - 2b \leq -1 \vee b - 2c < -1)$$

Štruktúra vety bude zachytená pomocou $F_{skel} = s_1 \wedge (s_2 \vee s_3)$, kde uvedené tri symboly reprezentujú tri atómy v poradí danom ich umiestnením vo vete.

Pri eliminácii premennej a je potrebné zobrať do úvahy dve podmienky: $-a+2c \leq -3$ (reprezentovanú symbolom s_1) a $2a-2b \leq -1$ (reprezentovanú pomocou s_2). Z týchto podmienok vznikne nová podmienka $-2b+4c \leq -7$, ktorá bude reprezentovaná symbolom s_4.

V zozname atómov ostali dve podmienky $b - 2c < -1$ (reprezentovaná s_3) a $-2b + 4c \leq -7$ (reprezentovaná s_4) nad dvomi premennými. Pri eliminácii premennej b vznikne nová podmienka $7 < -2$. Pretože sa jedná o nesplniteľnú podmienku, tak ju namiesto novým symbolom budeme reprezentovať priamo hodnotou $FALSE$.

Keďže v zozname atómov už nezostala žiadna podmienka, tak výsledná podoba F_{ohr} bude

$$(s_1 \wedge s_2 \rightarrow s_4) \wedge (s_4 \wedge s_3 \rightarrow FALSE)$$

reprezentujúca dve novo vytvorené podmienky. •

REDUKCIA POČTU GENEROVANÝCH PODMIENOK

Postup generovania nových podmienok je niekedy redundantný – generujú sa aj také podmienky, ktoré nie sú potrebné. Príčinou je to, že pri generovaní nových podmienok sa neberie do úvahy štruktúra pôvodnej vety, ktorá je zachytená v F_{skel}.

Ak by pôvodná veta mala tvar DNF (disjunktívnej normálnej formy), tak by bola reprezentovaná ako disjunkcia konjunktívnych klauzúl, kde každá z klauzúl je vyjadrením potenciálneho výrokového modelu vety. Potom má zmysel uvažovať vzájomné vzťahy iba tých podmienok, ktoré sú súčasťou rovnakej klauzuly. Ak dve podmienky nie sú spolu uvažované v rámci rovnakej klauzuly, tak nie je potrebné skúmať ich vzájomý vplyv.

Využiť takýto spôsob redukcie počtu generovaných podmienok je možné dvojakým spôsobom:

- transformovať vstupnú vetu na tvar DNF a následne osobitne riešiť $\mathcal{T}$-splniteľnosť pre každú z klauzúl osobitne,
- odhadovať príslušnosť podmienok k rovnakej klauzule bez realizácie transformácie vety na DNF.

Rozšírenie: SMT-lib formát

SMT-LIB FORMÁT

Samostatné SMT solvery ako špecifikačný jazyk pre definovanie úloh a interakciu so solverom používajú SMT-lib štandard. V ďalšom je uvedená malá podmnožina tohto jazyka, ktorá je však dostatočná pre vytváranie logických viet v tvare kombinácie výrokovej logiky s modulovou teóriou lineárnej aritmetiky. Jej možnosti sú ilustrované na nasledujúcom (samovysvetľujúcom) príklade špecifikácie.

```
(declare-const p Bool)
(declare-const q Bool)
(declare-const r Bool)
(declare-const x Int)
(declare-const y Int)
(declare-const s Real)
(declare-const t Real)

(assert (or (xor (xor p q) r)
            (and (not p) q r)
            (and p (not q) r)
            (and p q (not r))))
(assert (and (=> p q) (=> q r) (=> r p)))

(assert (= (* 2 x) (+ (* 4 y) 2)))
(assert (or (> x 5) (< 0 (ite (< x y) x y) )))

(assert (or (not (= p q))
            (and (>= t (- s 3)) (<= (* 2 t) (- s 1)))))

(check-sat)
(get-model)
```

Na začiatku sú deklarované použité symboly a premenné spolu s ich typom. Za týmito dekláráciami nasleduje zápis jednotlivých podmienok pomocou `assert` formy (sekvencia takýchto foriem je chápaná v zmysle ich konjunkcie). Na konci sú dva príkazy pre solver – pre kontrolu splniteľnosti (vrátane $\mathcal{T}$-splniteľnosti) a následne pre výpis modelu v prípade, že definície sú splniteľné.

Pre zápis podmienok je možné použiť logické operátory (*not*, *and*, *or*, =>, =, *xor*), aritmetické operátory (+, −, ∗, / pre reálne a *div* pre celočíselné delenie), relačné operátory (<, <=, >=, >, =) a *ite*.

Príklad 4.10 V predchádzajúcom príklade bola štruktúra vety reprezentovaná pomocou $F_{skel} = s_1 \wedge (s_2 \vee s_3)$ čo je možné upraviť na DNF tvar $F_{skel} = (s_1 \wedge s_2) \vee (s_1 \wedge s_3)$. Toto zodpovedá vete s dvomi konjunktívnymi klauzulami v tvare

$$(-a + 2c \leq -3 \wedge 2a - 2b \leq -1) \vee (-a + 2c \leq -3 \wedge b - 2c < -1)$$

Pri eliminácii premennej a z prvej klauzuly je opäť vygenerovaná implikácia $s_1 \wedge s_2 \rightarrow s_4$ kde s_4 reprezentuje nový atóm $-2b + 4c \leq -7$. Keďže podmienka reprezentovaná s_4 vznikla na základe podmienok prvej klauzuly, tak aj s_4 sa viaže na prvú klauzulu – a teda nie je potrebné porovnávať s_4 a s_3, pretože nepatria k rovnakej klauzule. Následkom toho implikácia

$$s_4 \wedge s_3 \rightarrow FALSE$$

nebude generovaná. •

Cvičenia

1. Pomocou metódy priameho kódovania rozdielovej logiky nájdite riešenie, ktoré spĺňa

   $$(a-c \geq 5 \vee c-a \geq 5) \wedge (b-a \geq 5 \vee a-b \geq 10) \wedge (d-c \geq 5 \vee c-d \geq 10)$$

   pre štyri reálne číselné premenné a, b, c a d, ktorých hodnoty sú obmedzené pomocou intervalov $0 \leq a, c \leq 12$ a $0 \leq b, d \leq 7$.

2. Pre konjunkciu lineárnych ohraničení

   $$-x - 3y \leq -12 \wedge x + y \leq 10 \wedge -x + y \leq -7$$

   definovaných nad reálnymi premennými $x \geq 0$ a $-\infty \leq y \leq \infty$ overte splniteľnosť danej konjunkcie a nájdite vhodné priradenie premenných, spĺňajúce dané ohraničenia. Riešte použitím

   - eliminácie premenných,
   - simplexovej metódy.

   Čo sa zmení pridaním ďalšieho lineárneho ohraničenia $x - 3y \leq 3$ a čo pridaním $-x + 3y \leq -3$?

3. Úlohu rozvrhu prípravy jedál z ilustračného príkladu na strane 98 riešte použitím prístupu

- včasnej kontroly (s použitím SAT solvera),
- neskorej kontroly (s použitím SMT solvera).

Pokúste sa nájsť riešenie s najkratšou celkovou dĺžkou. Pokúste sa nájsť riešenie s najkratšími prestojmi medzi operáciami v rámci receptov.

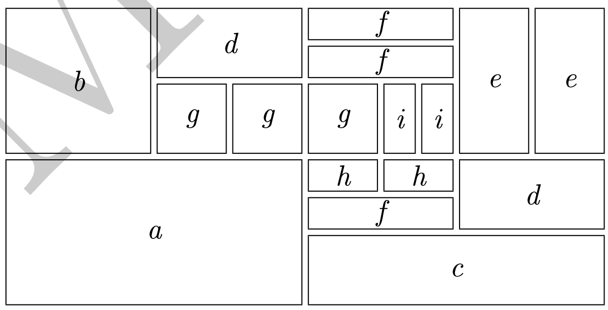
Skúste doplniť ďalšie podmienky:

- Šúľance so strúhankou nahraďte šúľancami s makom (pribudne mletie maku na mlynčeku v dĺžke 5 minút, ktoré môže byť v danom recepte zaradené na ľubovoľné miesto).
- Všetky recepty skončia približne v rovnakom čase (uvažujte ich ukončenie v rámci 10-minútového časového okna).

4. Vytvorte hlavolam Sudoku obsahujúci 81 (9x9) políčok, pričom v niektorých políčkach je umiestnená číslica z {1, 2, 3, 4, 5, 6, 7, 8, 9} zatiaľ čo iné políčka sú prázdne. Vygenerovaný hlavolam musí

   - mať práve jeden spôsob ako je možné doplniť číslice do prázdnych políčok tak, aby v žiadnom riadku, stĺpci a ani v žiadnom z deviatich 3x3 štvorcov neboli dve rovnaké číslice,
   - byť minimálny – nie je možné v ňom zameniť políčko s číslicou za prázdne políčko bez toho, aby sa nezvýšil počet možných doplnení.

5. Je definovaná sada štvoruholníkových objektov podľa nasledujúceho obrázku



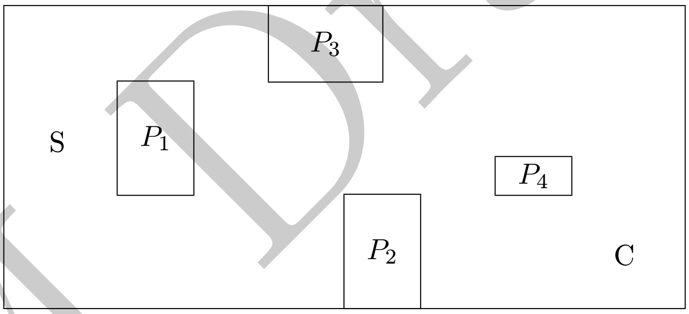
pozostávajúca zo 17 objektov 9 typov: typ a (s rozmermi 8 x 4), b (4 x 4), c (8 x 2), d (4 x 2), e (2 x 4), f (4 x 1), g (2 x 2), h (2 x 1) a i (1 x 2). Tieto objekty je potrebné vyrezať z tabule dlhej 17 jednotiek a širokej 9 jednotiek. Umiestnite dané objekty na tabuľu tak, aby sa navzájom neprekrývali a aby medzi každou dvojicou objektov bola medzera aspoň 0.1 jednotky.

6. Overte správnosť činnosti algoritmov pre určenie najväčšieho spoločného deliteľa dvoch kladných celých čísel. Uvažujte

   - Euklidov algoritmus,
   - Steinov binárny algoritmus.

   Overenie realizujte prístupom ohraničenej kontroly modelu – pri ohraničení počtu zmien vnútorného stavu algoritmu. Testujte, či algoritmus pre nejaké dve čísla a ich najväčšieho spoločného deliteľa dokáže tohto deliteľa potvrdiť .

7. V priestore sú uložené obdĺžnikové prekážky P_1 a P_2 (obe 10x15), P_3 (15x10) a P_4 (10x5) podľa nasledujúceho obrázku – ich pozícia daná ľavým dolným rohom je (15,15) pre prekážku P_1, (35,30) pre P_3, (45,0) pre P_2 a (65,15) pre P_4.



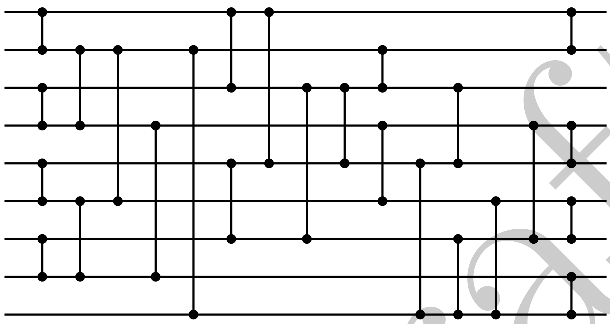
   Naplánujte trajektóriu zo štartovacej pozície S (5,20) po cieľovú pozíciu C (80,5), ktorá pozostáva z lineárnych segmentov.

8. Pri rozvrhovaní typu "j x m job shop" je zadaných j úloh, z ktorých každá pozostáva z m operácií vykonávaných na m strojoch. Poradie jednotlivých operácií a dĺžka operácií na jednotlivých strojoch závisí od úlohy. Nasledujúca tabuľka

úloha 1:	3,1	1,3	2,6	4,7	6,3	5,6
úloha 2:	2,8	3,5	5,10	6,10	1,10	4,4
úloha 3:	3,5	4,4	6,8	1,9	2,1	5,7
úloha 4:	2,5	1,5	3,5	4,3	5,8	6,9
úloha 5:	3,9	2,3	5,5	6,4	1,3	4,1
úloha 6:	2,3	4,3	6,9	1,10	5,4	3,1

je 6x6 benchmarkom (podľa Mutha a Thompsona). Hovorí, že prvá úloha najprv používa stroj 3 po dobu 1 časovej jednotky, potom stroj 1 po dobu 3 jednotiek, potom stroj 2 atď.

9. Na nasledujúcom obrázku je znázornená zotrieďovacia sieť, ktorá hodnoty na svojom (ľavom) vstupe poskytne v zotriedenom poradí na svojom (pravom) výstupe.



Jednotlivé prepojenia reprezentujú porovnanie dvoch hodnôt a ich prípadné prehodenie v prípade, ak sú v zlom poradí. Overte správnosť činnosti tejto siete.

10. Predpokadajte, že je daný orientovaný graf, ktorý je reprezentovaný pomocou výrokových symbolov typu $X_{i,j}$ – symbol je pravdivý vtedy, ak existuje orientovaná hrana z uzla i do uzla j, zatiaľ čo je nepravdivý v prípade, že takáto hrana neexistuje.

    Úlohou je určiť, či v danom grafe existuje slučka – cesta pozdĺž hrán, ktorá sa vráti do uzla z ktorého začínala.

Literatúra

1. Barrett, C. – Fontaine, P. – Tinelli, C.: The Satisfiability Modulo Theories Library (SMT-LIB), www.SMT-LIB.org. 2016.
2. Barrett, C. – Sebastiani, R. – Seshia, S.A. – Tinelli, C.: Satisfiability modulo theories. In: Handbook of satisfiability, Biere et al. (eds.). IOS Press, Amsterdam, 2009, 825–885.
3. De Moura, L. – Bjorner, N.: Satisfiability Modulo Theories: Introduction and Applications. Communication of the ACM, vol. 54, 2001, no. 9, 69–77.

4. Kroening, D. – Strichman, O.: Decision procedures (An algorithmic point of view). Springer, Berlin Heidelberg, 2008.

5. Monniaux, D.: A survey of Satisfiability Modulo Theory. In: Computer Algebra in Scientific computing, Springer, LNCS 9890, 2016, 401–425.

6. Nieuwenhuis, R. – Oliveras, A. – Tinelli, C.: Solving SAT and SAT Modulo Theories: From an abstract Davis-Putnam-Logemann-Loveland procedure to DPLL(T). Journal of the ACM, vol. 53, 2006, no. 6, 937–977.

7. Seshia, S.A.: Adaptive eager boolean encoding for arithmetic reasoning in verification. PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, 2005.

8. Strichman, O.: On Solving Presburger and Linear Arithmetic with SAT. In: Formal Methods in Computer-Aided Design, Springer, LNCS 2517, 2002, 160–170.

9. Yurichev, D.: SAT/SMT by Example,
https://sat-smt.codes/SAT_SMT_by_example.pdf