= OPTIONAL HOMEWORK =

~ Modelling reasoning ~

* Propositional logic

① (TRUTH TABLE)

Our friend is throwing a party. However, not everyone is getting along with all the others, so there are certain conditions imposed by the participants.

R_1. If John comes to the party, then so will Mary.

R_2. If Mary and Corey come to the party, then Dave also comes.

R_3. If Dave comes, then John doesn't come.

R_4. If Corey comes, then John also comes.

R_5. If John comes, then Corey or Dave come.

R_6. Mary doesn't come if Corey doesn't come.

R_7. Corey comes to the party if neither Mary nor John come to the party.

The party's organiser wants to know if it's even possible to satisfy all of these conditions.

$R_1.\ J \to M \equiv \neg J \vee M$

$R_2.\ M \wedge C \to D \equiv \neg M \vee \neg C \vee D$

$R_3.\ D \to \neg J \equiv \neg D \vee \neg J$

$R_4.\ C \to J \equiv \neg C \vee J$

$R_5.\ J \to C \vee D \equiv \neg J \vee C \vee D$

$R_6.\ \neg C \to \neg M \equiv C \vee \neg M$

$R_7.\ \neg M \wedge \neg J \to C \equiv M \vee J \vee C$

J – John comes to the party

M – Mary comes to the party

D – Dave comes to the party

C – Corey comes to the party

$U = R_1 \wedge R_2 \wedge R_3 \wedge R_4 \wedge R_5 \wedge R_6 \wedge R_7$

	J	M	C	D	R_1	R_2	R_3	R_4	R_5	R_6	R_7	U
i_1	T	T	T	T	T	T	F	T	T	T	T	F
i_2	T	T	T	F	T	F	T	T	T	T	T	F
i_3	T	T	F	T	T	T	F	T	T	F	T	F
i_4	T	T	F	F	T	T	T	T	F	F	T	F
i_5	T	F	T	T	F	T	F	T	T	T	T	F
i_6	T	F	T	F	F	T	T	T	T	T	T	F
i_7	T	F	F	T	F	T	F	T	T	T	T	F
i_8	T	F	F	F	F	T	T	T	F	T	T	F
i_9	F	T	T	T	T	T	T	F	T	T	T	F
i_{10}	F	T	T	F	T	F	T	F	T	T	T	F
i_{11}	F	T	F	T	T	T	T	T	T	T	T	F
i_{12}	F	T	F	F	T	T	T	T	T	F	T	F
i_{13}	F	F	T	T	T	T	T	F	T	T	T	F
i_{14}	F	F	T	F	T	T	T	F	T	T	T	F
i_{15}	F	F	F	T	T	T	T	T	T	T	F	F
i_{16}	F	F	F	F	T	T	T	T	T	T	F	F

The column U contains only the truth value F, so U is false in all 16 interpretations (all interpretations are anti-models) $\Rightarrow$ U is an inconsistent formula.

The conditions are contradictory, they cannot be all satisfied at the same time.

② (DEFINITION OF DEDUCTION)

The students of group 917 have to create a software aplication for their Fundamentals of Programming course. The most important feature of this app is Feature A, and the students want to know if they will be able to implement it, given the time pressure and the cost of this app.

H_1: Feature A will be implemented if feature B will be implemented and feature C will not be implemented.

H_2: If feature D will be implemented, then feature B will also be implemented.

H_3: If feature D takes less than a week to implement, then it will be implemented.

H_4: Feature C is too expensive and will not be implemented.

H_5: Feature D takes 4 days to implement.

C: Will feature A be implemented?

$f_1 = H_1 : B \wedge \neg C \to A$ (hypothesis)

$f_2 = H_2 : D \to B$ (hypothesis)

$f_3 = H_3 : D_W \to D$ (hypothesis)

$f_4 = H_4 : C_e \wedge \neg C$ (hypothesis)

$f_5 = H_5 : D_W$ (hypothesis)

$C : A$

A – feature A will be implemented

B – feature B will be implemented

C – feature C will be implemented

D – feature D will be implemented

D_W – feature D takes less than a week

C_e – feature C is too expensive

The deduction process:

$f_5, f_3 \vdash_{mp} D : f_6$ (modus ponens applied)

$f_6, f_2 \vdash_{mp} B : f_7$ (modus ponens applied)

$f_4 \vdash \neg C : f_8$ (simplification)

$f_7, f_8 \vdash B \wedge \neg C : f_9$ (conjunction in conclusions)

$f_9, f_1 \vdash_{mp} A : f_{10} = C$ (modus ponens applied)

The sequence $(f_1, f_2, f_3, \dots, f_9)$ is the deduction of C

$\Rightarrow$ Feature A will be implemented.

③ (SEMANTIC TABLEAUX)

Consider the hypotheses:

H_1. John will go to the pool if Alice and Michael will go.

H_2. If it is hot outside, then Alice will go to the pool.

H_3. It rained all week, but today it's really hot.

H_4. Michael will go to the pool only if his best friend, David, will go.

H_5. David is free and will go to the pool.

C: Will John go to the pool?

H_1: $A \wedge M \rightarrow J$

H_2: $HO \rightarrow A$

H_3: $R \wedge HO$

H_4: $D \rightarrow M$

H_5: $Df \wedge D$

C: J

A - Alice will go to the pool

M - Michael will go to the pool

J - John will go to the pool

D - David will go to the pool

HO - Hot outside

R - It has rained

Df - David is free

We know: $H_1, H_2, H_3, H_4, H_5 \models C$ if and only if $H_1 \wedge H_2 \wedge H_3 \wedge H_4 \wedge H_5 \wedge \neg C$ has a closed semantic tableau.
(Theorem of soundness and completness)

$(A \wedge M \to J) \wedge (HO \to A) \wedge R \wedge HO \wedge (D \to M) \wedge Df \wedge D \wedge \neg J$ (1)

α rule for (1)

$A \wedge M \to J$ (2)

$HO \to A$ (3)

R

HO

$D \to M$ (4)

Df

D

$\neg J$

β rule for (4)

$\neg D$ — $\otimes$ closed branch

M

β rule for (2)

$\neg A \vee \neg M$ (5) | J $\otimes$ closed branch

β rule for (5)

$\neg A$ | $\neg M$ $\otimes$ closed branch

β rule for (3)

$\neg HO$ closed branch $\otimes$ | A $\otimes$ closed branch

We have obtained a complete and closed tableau. We have 5 closed branches containing the following pairs of opposite literals: $(D, \neg D)$, $(\neg J, J)$, $(M, \neg M)$, $(HO, \neg HO)$, $(\neg A, A)$.

Therefore $H_1, H_2, H_3, H_4, H_5 \models C$.
"John will go to the pool."

④ (LOCK RESOLUTION)

Consider the hypotheses:

H_1. If Dave and Paul come outside, then we'll play football.

H_2. Dave comes outside if he finishes his logic and algebra homework.

H_3. Dave finishes his logic homework if he is helped by John.

H_4. John decided to to help Dave this time.

H_5. Dave finishes his algebra homework if today is Saturday.

H_6. Yesterday was Friday.

H_7. Paul comes outside no matter what.

C: Will we play football?

$H_1: D \land P \to Fb \equiv \neg D \lor \neg P \lor Fb : C_1$

$H_2: Lh \land Ah \to D \equiv \neg Lh \lor \neg Ah \lor D : C_2$

$H_3: Jh \to Lh \equiv \neg Jh \lor Lh : C_3$

$H_4: Jh : C_4$

$H_5: Sat \to Ah \equiv \neg Sat \lor Ah : C_5$

$H_6: Sat : C_6$

$H_7: P : C_7$

$C: Fb$

$\neg C : \neg Fb : C_8$

D - Dave comes outside

P - Paul comes outside

Fb - We'll play football

Lh - Dave finishes his logic homework,

Ah - Dave finishes his algebra homework

Jh - John helps (Dave)

Sat - Today is Saturday

First, we'll index each literal from the set of clauses (arbitrarly) and then, using lock resolution, we'll see if we can derive the empty clause $\square$ from the set $S = \{C_1, C_2, \dots, C_8\}$.

$C_1 = {}_{(3)}\neg D \vee {}_{(2)}\neg P \vee {}_{(1)}Fl$; $C_2 = {}_{(4)}\neg Lh \vee {}_{(5)}\neg Ah \vee {}_{(6)}D$

$C_3 = {}_{(7)}\neg Jh \vee {}_{(8)}Lh$; $C_4 = {}_{(9)}Jh$; $C_5 = {}_{(10)}\neg Sat \vee {}_{(11)}Ah$

$C_6 = {}_{(12)}Sat$; $C_7 = {}_{(13)}P$; $C_8 = {}_{(14)}\neg Fl$

$C_9 = Res^{lock}_{Jh}(C_3, C_4) = {}_{(8)}Lh$

$C_{10} = Res^{lock}_{Lh}(C_2, C_9) = {}_{(5)}\neg Ah \vee {}_{(6)}D$

$C_{11} = Res^{lock}_{Sat}(C_5, C_6) = {}_{(11)}Ah$

$C_{12} = Res^{lock}_{Ah}(C_{10}, C_{11}) = {}_{(6)}D$

$C_{13} = Res^{lock}_{Fl}(C_1, C_8) = {}_{(3)}\neg D \vee {}_{(2)}\neg P$

$C_{14} = Res^{lock}_{P}(C_{13}, C_7) = {}_{(3)}\neg D$

$C_{15} = Res^{lock}_{D}(C_{12}, C_{14}) = \square$

$\Rightarrow$ S is inconsistent $\Rightarrow$ C is deducible from the hypotheses, therefore: "We will play football."

⑤ (RESOLUTION - SET OF SUPPORT STRATEGY)

Consider the hypotheses:

H_1. If it is snowy, then we will stay inside.

H_2. If we'll stay inside and John will be with us, then we will play monopoly.

H_3. It has snowed all week and it's really cold.

H_4. John got back from his vacation yesterday.

H_5. If John is in town, then he will join us.

C. Will we play monopoly?

$H_1 : S \to I \equiv \neg S \vee I : C_1$

$H_2 : I \wedge J \to M \equiv \neg I \vee \neg J \vee M : C_2$

$H_3 : S \wedge C : C_3 \wedge C_4$

$H_4 : \mathcal{J}t : C_5$

$H_5 : \mathcal{J}t \to J \equiv \neg \mathcal{J}t \vee J : C_6$

$C : M$

$\neg C : \neg M : C_7$

$X = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7\}$

S - It's snowy

I - we stay inside

J - John will be with us

M - We play monopoly

C - It's cold

$\mathcal{J}t$ - John is in town

$X = \{C_1, C_2, C_3, C_4, C_5, C_6, C_7\}$

$\underline{\underline{C_1}} = \neg S \vee I$; $\underline{\underline{C_2}} = \neg I \vee \neg Y \vee M$; $\underline{\underline{C_3}} = S$; $\underline{\underline{C_4}} = C$;

$\underline{\underline{C_5}} = H$; $\underline{\underline{C_6}} = \neg H \vee Y$; $\underline{\underline{C_7}} = \neg M$

C is a pure literal in $X \Rightarrow C_4$ is eliminated. We obtain $X' = \{C_1, C_2, C_3, C_5, C_6, C_7\}$. The set of support of X' is $Y = \{C_7\}$ (the negation of the conclusion).

We will derive $\square$ from X':

$C_8 = Res(C_2, C_7) = \neg I \vee \neg Y$, $\quad Y = \{C_7, C_8\}$

$C_9 = Res(C_8, C_6) = \neg I \vee \neg H$, $\quad Y = \{C_7, C_8, C_9\}$

$C_{10} = Res(C_9, C_5) = \neg I$, $\quad Y = \{C_7, C_8, C_9, C_{10}\}$

$C_{11} = Res(C_{10}, C_1) = \neg S$, $\quad Y = \{C_7, C_8, C_9, C_{10}, C_{11}\}$

$C_{12} = Res(C_{11}, C_3) = \square$

$CNF(H_1 \wedge H_2 \wedge H_3 \wedge H_4 \wedge H_5 \wedge \neg C) \vdash_{Res} \square \Rightarrow C$ is deducible from the hypotheses, therefore "We will play monopoly"

⊛ Predicate logic

① (CLOCK RESOLUTION)

H_1. If a is b's boss and a is horrible, then b will be sad.

H_2. If x is y's boss, x retires, and z is x's oldest son, then z will become y's boss.

H_3. If t reaches a top position even tough he is not a hard worker, then he will behave badly with others.

H_4. All people who behave badly with others are horrible.

H_5. David is John's boss.

H_6. Michael is David's oldest son.

H_7. David retired.

H_8. Michael is not a hard worker.

C. John will be sad.

$H_1: (\forall a)(\forall b)(boss(a,b) \wedge horr(a) \rightarrow sad(b))$

$H_2: (\forall x)(\forall y)(\forall z)(boss(x,y) \wedge retires(x) \wedge oldest\_son(x,z) \longrightarrow$
$boss(z,y))$

$H_3: (\forall t)(\forall m)(boss(t,m) \wedge \neg hard\_working(t) \rightarrow behave\_bad(t))$

$H_4: (\forall s)(behave\_bad(s) \rightarrow horr(s))$

$H_5:$ boss(David, John)

$H_6:$ oldest_son(David, Michael)

$H_7:$ retires(David)

$H_8:$ $\neg$ hard_working(Michael)

C: sad(John)

Where x, y, z, t, s, m, a, b variables
David, John, Michael - constants
horr, sad, retires, behave_bad - unary predicates
boss, oldest_son - binary predicates

The clausal normal forms corresponding to the hypotheses and the negation of the conclusion are:

$(H_1)^c$: $\neg boss(a,b) \vee \neg horr(a) \vee sad(b) = C_1$

$(H_2)^c$: $\neg boss(x,y) \vee \neg retires(x) \vee \neg oldest\_son(x,z) \vee boss(z,y) = C_2$

$(H_3)^c$: $\neg boss(t,m) \vee hard\_working(t) \vee behave\_bad(t) = C_3$

$(H_4)^c$: $\neg behave\_bad(s) \vee horr(s) = C_4$

$(H_5)^c$: $boss(David, John) = C_5$

$(H_6)^c$: $oldest\_son(David, Michael) = C_6$

$(H_7)^c$: $retires(David) = C_7$

$(H_8)^c$: $\neg hard\_working(Michael) = C_8$

$(\neg C)^c$: $\neg sad(John) = C_9$

We apply lock resolution to the set of clauses

$S = \{C_1, C_2, \dots, C_9\}$

$C_1 = \underset{(1)}{\neg} boss(a,b) \lor \underset{(2)}{\neg} horr(a) \lor \underset{(3)}{} sad(b)$

$C_2 = \underset{(4)}{\neg} boss(x,y) \lor \underset{(5)}{\neg} retires(x) \lor \underset{(6)}{\neg} oldest\_son(x,z) \lor \underset{(7)}{} boss(z,y)$

$C_3 = \underset{(8)}{\neg} boss(t,m) \lor \underset{(9)}{} hard\_working(t) \lor \underset{(10)}{} behave\_bad(t)$

$C_4 = \underset{(11)}{\neg} behave\_bad(s) \lor \underset{(12)}{} horr(s)$

$C_5 = \underset{(13)}{} boss(David, John)$

$C_6 = \underset{(14)}{} oldest\_son(David, Michael)$

$C_7 = \underset{(15)}{} retires(David)$

$C_8 = \underset{(16)}{\neg} hard\_working(Michael)$

$C_9 = \underset{(17)}{\neg} sad(John)$

$C_{10} = Res^{Pr}_{[x \leftarrow David,\, y \leftarrow John]}(C_2, C_5) = \underset{(5)}{\neg} retires(David) \lor \underset{(6)}{\neg} oldest\_son(David, z) \lor \underset{(7)}{} boss(z, John)$

$C_{11} = Res^{Pr}(C_7, C_{10}) = \underset{(6)}{\neg} oldest\_son(David, z) \lor \underset{(7)}{} boss(z, John)$

$C_{12} = Res^{Pr}_{[z \leftarrow Michael]}(C_6, C_{11}) = \underset{(7)}{} boss(Michael, John)$

$$C_{13} = Res^{Pr}_{[t \leftarrow Michael,\, m \leftarrow John]}(C_{12}, C_3) =_{(9)} hard\_working(Michael) \vee_{(10)} behave\_bad(Michael)$$

$$C_{14} = Res^{Pr}(C_{13}, C_8) = behave\_bad(Michael)$$

$$C_{15} = Res^{Pr}_{[s \leftarrow Michael]}(C_{14}, C_4) =_{(12)} horr(Michael)$$

$$C_{16} = Res^{Pr}_{[e \leftarrow Michael,\, h \leftarrow John]}(C_1, C_{12}) =_{(2)} \neg horr(Michael) \vee_{(3)} sad(John)$$

$$C_{17} = Res^{Pr}(C_{15}, C_{16}) = sad(John)$$

$$C_{18} = Res^{Pr}(C_9, C_{17}) = \square$$

The most general unifier:

$[x \leftarrow David, y \leftarrow John, z \leftarrow Michael, t \leftarrow Michael, m \leftarrow John, s \leftarrow Michael, e \leftarrow Michael, h \leftarrow John]$

$$S \vdash^{Pr}_{Res} \square \Rightarrow S \text{ is inconsistent}$$

$\Rightarrow$ The deduction $H_1, H_2, H_3, H_4, H_5, H_6, H_7, H_8 \vdash C$ holds.

② (LINEAR RESOLUTION)

H_1. All high-level programming languages are intuitive.

H_2. Python is a high-level programming languages.

H_3. No programming language is boring.

H_4. Programming languages that are intuitive and interesting are a pleasure to be learned.

C. Python is a pleasure to be learned.

$H_1: (\forall x)(high\_level(x) \rightarrow intuitive(x))$

$H_2: high\_level(Python)$

$H_3: \neg(\exists y)(boring(y)) \equiv (\forall y)(\neg boring(y))$

$H_4: (\forall z)(intuitive(z) \wedge \neg boring(z) \rightarrow pleasure(z))$

$C: pleasure(Python)$

, where x, y, z – variables

Python – constant

high_level, intuitive, boring, pleasure – unary predicates

The hypotheses and the negation of the conclusion in clausal normal form are:

$(H_1)^c = \neg\, \text{high-level}(x) \vee \text{intuitive}(x) = C_1$

$(H_2)^c = \text{high-level}(\text{Python}) = C_2$

$(H_3)^c = \neg\, \text{boring}(y) = C_3$

$(H_4)^c = \neg\, \text{intuitive}(z) \vee \text{boring}(z) \vee \text{pleasure}(z) = C_4$

$(\neg C)^c = \neg\, \text{pleasure}(\text{Python}) = C_5$

We apply linear resolution to the set $S = \{C_1, C_2, C_3, C_4, C_5\}$ with C_1 as the top clause.

$$
\begin{tikzcd}
\boxed{C_1} \arrow[d] & \boxed{C_2} \arrow[dl, "{[x \leftarrow \text{Python}]}"] \\
\boxed{C_6 = \text{intuitive}(\text{Python})} \arrow[d] & \boxed{C_4} \arrow[dl, "{[z \leftarrow \text{Python}]}"] \\
\boxed{C_7 = \text{boring}(\text{Python}) \vee \text{pleasure}(\text{Python})} \arrow[d] & \boxed{C_5} \arrow[dl] \\
\boxed{C_8 = \text{boring}(\text{Python})} \arrow[d] & \boxed{C_3} \arrow[dl, "{[y \leftarrow \text{Python}]}"] \\
\boxed{C_9 = \square} &
\end{tikzcd}
$$

$$C_6 = Res^{Pr}_{[x \leftarrow Python]}(C_1, C_2) = intuitive(Python)$$

$$C_7 = Res^{Pr}_{[z \leftarrow Python]}(C_6, C_4) = boring(Python) \vee pleasure(Python)$$

$$C_8 = Res^{Pr}(C_7, C_5) = boring(Python)$$

$$C_9 = Res^{Pr}_{[y \leftarrow Python]}(C_3, C_8) = \square$$

The most general unifier generated is:
$[x \leftarrow Python, y \leftarrow Python, z \leftarrow Python]$

$S \vdash^{lin}_{Res} \square \Rightarrow S$ is an inconsistent set and we conclude that:

"Python is a pleasure to be learned."

③ (GENERAL RESOLUTION)

H_1. A smart person person will solve real-life problems.
H_2. If a person studies hard, then he/she will be a smart person.
H_3. Someone who solves real-life problems is appreciated.
H_4. John studies hard.
C. John is appreciated

H_1 : $(\forall x)(smart(x) \rightarrow solves\_problems(x))$

H_2 : $(\forall y)(study(y) \rightarrow smart(y))$

H_3 : $(\forall z)(solves\_problems(z) \rightarrow appreciated(z))$

H_4 : $study(John)$

C : $appreciated(John)$

, where x, y, z - variables
John - constant
smart, solves_problems, study, appreciated - unary predicates

The clausal normal forms of the hypotheses and the negation of the conclusion are as follows:

$(H_1)^c$: $\neg$ smart(x) $\vee$ solves_problems(x) $= C_1$

$(H_2)^c$: $\neg$ study (y) $\vee$ smart(y) $= C_2$

$(H_3)^c$: $\neg$ solves_problems(z) $\vee$ appreciated (z) $= C_3$

$(H_4)^c$: study (John) $= C_4$

$(\neg C)^c$: $\neg$ appreciated (John) $= C_5$

We apply general predicate resolution to the set of clauses $S = \{C_1, C_2, C_3, C_4, C_5\}$

$C_6 = Res^{Pr}_{[y \leftarrow John]}(C_2, C_4) =$ smart (John)

$C_7 = Res^{Pr}_{[z \leftarrow John]}(C_3, C_5) = \neg$ solves_problems (John)

$C_8 = Res^{Pr}_{[x \leftarrow John]}(C_1, C_7) = \neg$ smart (John)

$C_9 = Res^{Pr}(C_6, C_8) = \square$

The most general unifier generated: [x←John, y←John, z←John]

$S \vdash^{Pr}_{Res} \square \Rightarrow S$ is inconsistent and the deduction $H_1, H_2, H_3, H_4 \vdash C$ holds.

"John is appreciated."

④ (SEMANTIC TABLEAUX)

H_1. All video games are interesting.

H_2. No large computer program is easy to make.

H_3. Computer programs that are easy to make are boring.

H_4. Tetris is a video game.

C. Tetris is a large computer program.

$H_1 : (\forall x)(vg(x) \to interesting(x))$

$H_2 : \neg(\exists x)(large(x) \to make\text{-}easy(x))$

$H_3 : (\forall x)(make\text{-}easy(x) \to \neg interesting(x))$

$H_4 : vg(Tetris)$

$C : large(Tetris)$

, where x - variables

Tetris - constant

vg, interesting, large, make-easy - unary predicates

$H_1, H_2, H_3, H_4 \models C$ if and only if

$H_1 \wedge H_2 \wedge H_3 \wedge \neg C$ has a closed semantic tableau.

(Theorem of soundness and completness)

$$(\forall x)(vg(x) \rightarrow interesting(x)) \land \neg(\exists x)(large(x) \rightarrow make\text{-}easy(x)) \land (\forall x)(make\text{-}easy(x) \rightarrow \neg interesting(x)) \land vg(Tetris) \land \neg large(Tetris) \quad (1)$$

α rule for (1)

$$(\forall x)(vg(x) \rightarrow interesting(x)) \quad (2)$$

$$\neg(\exists x)(large(x) \rightarrow make\text{-}easy(x)) \quad (3)$$

$$(\forall x)(make\text{-}easy(x) \rightarrow \neg interesting(x)) \quad (4)$$

$$vg(Tetris)$$

$$\neg large(Tetris)$$

γ rule for (2)

$$vg(Tetris) \rightarrow interesting(Tetris) \quad (5)$$

$(\forall x)(vg(x) \rightarrow interesting(x))$ - copy of formula (2)

| γ rule for (3)

$\neg(\text{large}(\text{Tetris}) \rightarrow \text{make\_easy}(\text{Tetris}))$ (6)

|

(4)

$\neg(\exists x)(\text{large}(x) \rightarrow \text{make\_easy}(x))$ - copy of formula (3)

| γ rule for (4)

$\text{make\_easy}(\text{Tetris}) \rightarrow \neg\text{interesting}(\text{Tetris})$ (7)

|

$(\forall x)(\text{make\_easy}(x) \rightarrow \neg\text{interesting}(x))$ - copy of formula (4)

| α rule for (6)

$\text{large}(\text{Tetris})$

|

$\neg\text{make\_easy}(\text{Tetris})$

⊗ closed branch

$H_1 \wedge H_2 \wedge H_3 \wedge H_4 \wedge \neg C$ has a closed semantic tableau containing the pairs of opposite literals ($\neg$ large(Tetris), large(Tetris)).

$\Rightarrow$ The deduction $H_1, H_2, H_3, H_4 \vdash C$ holds.

"Tetris is a large computer program."