ANATOLI DEGTYAREV AND ANDREI VORONKOV

KANGER'S CHOICES IN AUTOMATED REASONING

Automated deduction, or automated theorem proving is a branch of science that deals with automatic search for a proof. The contribution of Kanger to automated deduction is well-recognized. His monograph [1957] introduced a calculus **LC**, which was one of the first calculi intended for automated proof-search. His article [1963] was later republished as [Kanger 1983] in the collection of "classical papers on computational logic". Kanger's [1963] (and also [1959]) calculi used some interesting features that have not been noted for a number of years, and the importance of which in the area of automated deduction has been recognized only much later.

Kanger [1963] gives no proofs and uses very succinct presentation. Automated deduction is an area in which very subtle changes in definitions and assertions may lead to inconsistent conclusions. Kanger's [1963] area was theorem proving in sequent calculi with equality and function symbols. Most papers published in this area before 1995 contained serious mistakes, except for Kanger's.

Now, when we are equipped with the impressive amount of techniques developed in this area, we are amazed by the incredible intuition of Kanger that allowed him to choose elegant, interesting (and correct) solutions among many possible choices. This article explains these choices and their place in modern automated deduction.

1. $\models\ \equiv\ \vdash$

The title of this section $\models \equiv \vdash$ is the logo of the Association for Logic Programming: truth is equivalent to provability. The equivalence of validity and provability for classical logic was proved by Gödel [1930] and is known as Gödel's completeness theorem. The notions of truth and validity in logic are formulated as semantical properties, while the notion of provability is defined in a purely syntactical way, so there seems to be a gap between the two notions.

In 1955–1957 several new proofs of Gödel's completeness theorem appeared [Beth 1955, Hintikka 1955, Schütte 1956, Kanger 1957] in which model theory and proof theory were connected in a very natural manner. They

G. Holmström-Hintikka, S. Linström and R. Slivinski (eds.),
Collected Papers of Stig Kanger with Essays on his Life and Work, Vol. II, 53–67.

are based on the idea of searching for countermodels of a given formula F by applying a proof-search procedure to F (i.e. trying to establish $\vdash F$).

Kanger proposed to search for a proof in a *sequent calculus* named **LC** [Kanger 1957]. Cut-free sequent calculi for first-order logic have been introduced by Gentzen [1934]. They turned out to be an important tool for investigating basic proof-theoretic problems [e.g. Gentzen 1936, Girard 1987]. It has also been realized that sequent systems give a convenient tool for designing proof-search algorithms by using the rules of a calculus backwards (i.e. from the conclusion to the premise). To prove a sequent S *"we start from below with S and proceed upwards from level to level in the tree form. At each level the sequent of the next level above are uniquely and effectively determined — if there is such a level. If there is no such level, this fact is effectively determined, so that the process may brought to an end."* [Kanger 1957, page 31]. Consider some choices that arise when one formalizes sequent calculi.

Choice 1 (structure rules) In the original Gentzen's LK a sequent was an expression $\Gamma \rightarrow \Delta$, where Γ, Δ are *sequences* of formulas. Since Γ and Δ play the role of a conjunction and a disjunction, respectively, the logical semantics of a sequent is independent of the order of formulas in Γ, Δ. Neither does it depend on duplicate occurrences of formulas in Γ or Δ. Therefore, Gentzen had to introduce several *structure rules* that allow one to interchange and duplicate formulas in Γ, Δ, and also add new formulas:

$$\frac{\Gamma \rightarrow \Delta_1, B, A, \Delta_2}{\Gamma \rightarrow \Delta_1, A, B, \Delta_2} \qquad \frac{\Gamma_1, B, A, \Gamma_2 \rightarrow \Delta}{\Gamma_1, A, B, \Gamma_2 \rightarrow \Delta}$$

$$\frac{\Gamma \rightarrow \Delta_1, A, A}{\Gamma \rightarrow \Delta, A} \qquad \frac{\Gamma, A, A \rightarrow \Delta}{\Gamma, A \rightarrow \Delta}$$

$$\frac{\Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, A} \qquad \frac{\Gamma \rightarrow \Delta}{\Gamma, A \rightarrow \Delta}$$

These rules are called *exchange*, *contraction* and *weakening*. The use of these rules introduced unnecessary technical details in proofs of [Gentzen 1934]. In order to avoid complications, other structures than sequences should be adopted. One obvious choice is the use *sets* instead of sequents. This again makes the formalization of sequent calculi quite complex. Suppose that Γ, Δ are sets and consider the following rule of sequent calculi:

$$\frac{\Gamma \to \Delta \cup \{A\}}{\Gamma \to \Delta \cup \{A \vee B\}} (\to \vee)$$

Let Γ be empty and consider four different instantiations for Δ: $\{\}$, $\{A\}$, $\{A \vee B\}$, and $\{A, A \vee B\}$. We obtain the following four instances of this rule:

$$\frac{\to \{A\}}{\to \{A \vee B\}} (\to \vee) \qquad \frac{\to \{A\}}{\to \{A, A \vee B\}} (\to \vee)$$

$$\frac{\to \{A, A \vee B\}}{\to \{A \vee B\}} (\to \vee) \qquad \frac{\to \{A, A \vee B\}}{\to \{A, A \vee B\}} (\to \vee)$$

The last one is absurd, among all four instances only the first one is enough to preserve completeness. Therefore, if we choose sets, we have to impose several restrictions on the inference rules. If we prohibit A and $A \vee B$ occur in Δ, we may eventually loose completeness. Even if we impose no restrictions we might still be in need of the weakening rule. So what is the right choice for sequent and structure rules in sequent calculi?

Kanger's Choice 1 One distinctive feature of the calculi used in [Kanger 1957, Kanger 1963] is the *full absence of structure rules*. In order to achieve this, sequent are made of *multisets* of formulas and some rules are modified. The use of multisets eliminates the exchange rule. The use of contraction rule is replaced by the explicit duplication of formulas in some (but not all!) rules and changes in some other rules. For example, the $(\to\exists)$ rule in Kanger's system is

$$\frac{\Gamma \to \Delta, \exists x\varphi(x), \varphi(t)}{\Gamma \to \Delta, \exists x\varphi(x)} (\to \exists)$$

(the formula $\exists x\varphi(x)$ is explicitly duplicated), and the rule $(\to \vee)$ is changed into

$$\frac{\Gamma \to \Delta, A, B}{\Gamma \to \Delta, A \vee B} (\to \vee).$$

Finally, to get rid of weakening axioms $\Gamma, A \to \Delta, A$ are used instead of more traditional $A \to A$.

Completeness can be proved for virtually any variant of sequent calculi, but even completeness proofs meet small technical problems when it comes to structure rules. The choice made by Kanger to design a system without structure rules at all has now become de facto standard.

Kleene [1952] also described the sequent system G3 with invertible rules, but this property was realized straightforwardly by retaining the principal formula in the premise(s). Later Kleene's G3 was transformed to the system G4 [Kleene 1967][1], which was essentially the system **LC**.

Choice 2 (variants of rules) For some logical connectives, we have a choice among various sequent calculus rules. For example, for the proof disjunction one can use either the following two rules:

$$\frac{\Gamma \rightarrow \Delta, A}{\Gamma \rightarrow \Delta, A \vee B}\,(\rightarrow \vee) \quad \text{and} \quad \frac{\Gamma \rightarrow \Delta, B}{\Gamma \rightarrow \Delta, A \vee B}\,(\rightarrow \vee)$$

or just one rule

$$\frac{\Gamma \rightarrow \Delta, A, B}{\Gamma \rightarrow \Delta, A \vee B}\,(\rightarrow \vee)$$

The first choice seems to reflect the semantics of disjunction in a more intuitive way. Nevertheless, in Kanger's system the choice of the second rule is made. Why?

Kanger's Choice 2 The main answer is: all inference rules of Kanger's system are *invertible*. A rule is called invertible if the derivability of the conclusion implies the derivability of the premises. For automatic proof-search invertibility of rules is really a remarkable property. If a sequent $\mathcal{S}$ is unprovable, then *any* derivation tree for $\mathcal{S}$ has a branch containing a countermodel for $\mathcal{S}$. It allowed Kanger to prove completeness *"by means of arguments which are new in some respect and which involve a new turn to the notion of validity"* [Kanger 1957, page 7]. It also allows one to search for a proof in a "don't care" matter: after we have selected a rule to apply, there is no need to undo the selection.

Kleene [1967] notes that the use of his system G4 for proving the completeness theorem *"is quite close to Beth [1955] which gave the present writer the idea for it. In some respect it more resembles Kanger [1957], as the author learned after working it out."*

2. USE OF A SEQUENT CALCULUS AS A DECISION PROCEDURE FOR PREDICATE LOGIC

Kanger was one of the first who used a particular logical calculus as a decision procedure in the backward direction[2]. The point was to guarantee termination of the procedure on target classes of formulas. As examples Kanger considered the class of quantifier-free formulas and the class of $\forall^*\exists^*$ formulas (without functional symbols). Later, Wang [1960] described and implemented a procedure solving this class of formulas, also using backward proof-search in sequent calculi.

The possibility to obtain decision procedures for propositional logics, classical and intuitionistic, using backward proof-search in cut-free Gentzen type calculi was also noted by Kleene [1952]. Later, the use of derivations in machine-oriented calculi to decide some classes of predicate logic has become a generally accepted area of research [Maslov 1964, Kallick 1968, Maslov 1968, Joyner jr 1976, Fermüller, Leitsch, Tammet & Zamov 1993, Leitsch, Fermüller & Tammet 1999].

3. PROOF-SEARCH VIA LOGICAL CALCULI

As soon as the first programs for proving theorems in predicate logic appeared [Prawitz, Prawitz & Voghera 1960, Wang 1960, Gilmore 1960, Davis & Putnam 1960] it has become clear that the main problem consists in instantiating variables in the application of $(\rightarrow \exists)$ and $(\forall \rightarrow)$) rules (also called γ-rules due to [Smullyan 1968, Fitting 1996]).

$$\frac{\Gamma \rightarrow \Delta, \varphi[t/x], \exists x\varphi}{\Gamma \rightarrow \Delta, \exists x\varphi}(\rightarrow \exists) \quad \text{and} \quad \frac{\Gamma, \varphi[t/x], \forall x\varphi \rightarrow \Delta}{\Gamma, \forall x\varphi \rightarrow \Delta}(\rightarrow \forall)$$

Here sequent are represented by expressions of the form $\Gamma \rightarrow \Delta$, where Γ and Δ are multisets of formulas.

Choice 3 (variable instantiation in γ-rules) How to instantiate variables by terms in γ-rules? The early methods of automated reasoning used the so-called *level saturation*. The set of all variable-free terms was enumerated (usually respecting depending on the term depth) and terms have been substituted one by one in that order. However, it was clear that such a solution is far from the best.

Kanger's Choice 3 The system of Kanger used a new strategy for instantiating variables in the applications of γ-rules. His strategy of instantiating variables

http://www.springer.com/978-1-4020-0111-6

Collected Papers of Stig Kanger with Essays on his Life and Work Volume II
Holmström-Hintikka, G.; Lindström, S.; Sliwinski, R. (Eds.)
2001, XII, 281 p., Hardcover
ISBN: 978-1-4020-0111-6